



1 0 1 7 0 8 2 0 0 3 / U 0 3 8 6 8  
Rec'd PCT/PTO 0 2 MAR 2005



INVESTOR IN PEOPLE

The Patent Office  
Concept House  
Cardiff Road  
Newport  
South Wales  
NP10 8QQ

REC'D 24 OCT 2003

WIPO PCT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

**BEST AVAILABLE COPY**

Signed

*W. Evans*

Dated 24 September 2003

**PRIORITY  
DOCUMENT**

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



09/01/77 E746564-1 002246  
P01/7700 0.00-0220790.0

# Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road  
Newport  
South Wales  
NP10 8QQ

1. Your reference	P014968GB MJH		
2. Patent application number (The Patent Office will fill in this part)	06 SEP 2002	0220790.0	
3. Full name, address and postcode of the or of each applicant (underline all surnames)	CRESSET BIOMOLECULAR DISCOVERY LIMITED SPIRELLA BUILDING, SUITE 203 BRIDGE ROAD LETCHWORTH HERTFORDSHIRE, SG6 4ET  0846 069 3001 Patents ADP number (if you know it)  If the applicant is a corporate body, give the country/state of its incorporation UNITED KINGDOM		
4. Title of the invention	SEARCHABLE MOLECULAR DATABASE		
5. Name of your agent (if you have one)	D Young & Co		
"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)	21 New Fetter Lane London EC4A 1DA		
Patents ADP number (if you know it)	59006		
6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority application number (if you know it)	Date of filing (day / month / year)
	NONE		
7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing (day / month / year)	
8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if: a) any applicant named in part 3 is not an inventor, or b) there is an inventor who is not named as an applicant, or c) any named applicant is a corporate body. See note (d))	Yes		

Patents Form 1/77


9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form 0

Description 65

Claim(s) 8

Abstract 1

Drawing(s) 16 + 16 

10. If you are also filing any of the following, state how many against each item.

Priority documents 0

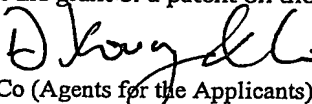
Translations of priority documents 0

Statement of inventorship and right to grant of a patent (Patents Form 7/77) 4

Request for preliminary examination and search (Patents Form 9/77) 1

Request for substantive examination (Patents Form 10/77) 1

Any other documents 0  
(please specify)

11. I/We request the grant of a patent on the basis of this application.  
Signature  Date 06.09.02  
D Young & Co (Agents for the Applicants)

12. Name and daytime telephone number of person to contact in the United Kingdom Dr Miles Haines 023 8071 9500

**Warning**

*After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.*

**Notes**

- a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.
- b) Write your answers in capital letters using black ink or you may type them.
- c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.
- d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.
- e) Once you have filled in the form you must remember to sign and date it.
- f) For details of the fee and ways to pay please contact the Patent Office.

TITLE OF THE INVENTION

SEARCHABLE MOLECULAR DATABASE

BACKGROUND OF THE INVENTION

5 The invention relates to a database of representations of molecules in different conformations which can be searched in order to find molecular conformations with similar field properties, as is useful for drug discovery.

10 A number of databases exist which allow comparison between structural representations of large numbers of molecules in different conformations [see e.g. references 1, 2]. Databases of this kind are useful for pharmaceutical research, since a known compound with a particular known activity can be used as a search query to identify other compounds with similar molecular structures. These other compounds can then be used as leads and can be studied to establish whether they exhibit similar activity.

15 One way to compare molecular conformations is to perform atom-atom searching in which each atom and bond of a molecule (including properties such as valence charge) is compared. Many algorithms have been produced to accomplish atom-atom comparison searching. A popular algorithm is that produced by Ullman or derivations based upon it. Whilst atom-atom searching is an effective way of comparing molecules, it is computationally intensive and hence slow. Search speeds become unacceptably slow for the average user even when searching across databases containing only a modest number of records.

25 To speed up the searching process it is conventional to initially perform an index-based search before atom-atom searching, which is then limited to the hits found in the index-based search. An index is a condensed representation of a

molecular conformation. A commonly used index type is the bit string (also referred to as a bit map). Bit strings can be rapidly compared using bit-wise operations.

5 For each molecular conformation an index is created from a definition of the conformation based on its structural properties, such as its atom types and properties of the inter-atomic bonds, such as bond length, angle etc. Two common bit string indexing methods use structural key indexes (also referred to as data dictionary indexes) and fingerprint indexes (also referred to as hashed indexes).

10 Much work has been carried on devising less specific representations for molecules. These take features of a molecule and reduce them to character representations, for example aromatic rings (A), linker chains (CH<sub>2</sub>) (L), electron withdrawing atoms (W), electron donating atoms (D), hydrogen acceptor atoms (HA), and hydrogen donating groups (HD). This allows a complex molecule to be  
15 represented by a simple abbreviated reduced molecule. These reduced molecules can be indexed just as if they had full atom representation, and used in search and metric calculations.

Through the use of similarity metrics researchers have devised clustering  
20 methods. These include K-Means, Nearest-Neighbour and Jarvis-Patrick algorithms, to name a few. These allow sets of bit strings to be grouped into bins or clusters, indicating that some relationship exists between them. Once clustered the bit strings may be further analysed to search for common bits (features) which tend to predominate in specific groups. These features have then been utilised further in  
25 quantitative structure-activity relationship (QSAR) analysis to relate biological activity with bit features. QSAR analysis is a standard term describing the calculation or measurement of one or more properties of a set of molecules and then attempting to relate the biological activities of the molecules to their properties (e.g. by regression).

While index-based searching across molecular databases has proved to be a powerful tool, it has some limitations. In particular, the searching is not generally good at finding new lead compounds which are structurally dissimilar to the search query compound. This is a consequence of the structure-based approach used in  
5 existing databases for the indexing. It is therefore desired to create a molecular database with an improved indexing system which is capable of finding lead compounds independent of structural similarity.

## SUMMARY OF THE INVENTION

Viewed from a first aspect the present invention provides a computer system comprising a database having a plurality of records, wherein each record comprises a  
5 field point representation representing field extrema for a conformation of a chemical structure.

Field point representations are independent of the structural class of a chemical structure. By providing a database with records comprising field point  
10 representations, searches can be performed by field point representation rather than chemical structure. Advantageously, searches can identify chemical structures of different structural class to that of a search query. Thus, the database can provide hits which are not be obtainable by known chemical structure databases and hits that are likely to have diverse chemical structures.

15

In a particular embodiment the database includes records for multiple conformations of the same chemical structure. Advantageously, multiple field point representations for the same chemical structure can be searched, increasing the likelihood of the chemical structure being included as a hit in the search results.

20

In one embodiment an index of the field point representation is associated with each record, the index being a searchable representation of the field point representation.

25

Preferably the index is a string. Each element of the string may be a binary digit (bit) so that the string is a bit string. Alternatively, the string elements may be more than two-valued, for example they may have values in the range 0 to 3 or 1 to 10. In this case the string elements are referred to as bins. (Use of bits for the string elements can thus be thought of as a special case in which the bin can only adopt two-

values.) Advantageously, by using a string, known string manipulation techniques can be used.

5 Multiple indexes of the field point representation may be associated with each record, the multiple indexes being representations of the field point representation at different precision levels. This enables a user to search at different precision levels.

In a preferred embodiment, the index is a string of length  $n$  and the computer system comprises an indexing mechanism for generating an index of a field point representation. The indexing mechanism is configured to:

- 10 (i) generate a numeric identifier from a characteristic of the field point representation;
- (ii) generate one or more numbers in a range from 1 to  $n$  (e.g. 0 to  $n-1$ ) in dependence on the numeric identifier;
- 15 (iii) incrementing the bins in the string that correspond to the one or more numbers; and
- (iv) optionally repeat (i) to (iii) for another characteristic of the field point representation.

20 Thus, a mechanism for generating a string from a field point representation is provided.

A characteristic of the field point representation may include one or more of:

- the number of field points of a particular field of the field point representation;
- 25 the particular field and energy of a field point in the field point representation; and
- the respective energies of and distance between a field point pairing in the field point representation.



In a preferred embodiment the indexing mechanism is configured to generate one or more numbers in a range from 1 to n in dependence on the numeric identifier by using a deterministic function, such as a pseudo-random number generator or a hash function.

5

The computer system may also comprise a searching mechanism configured to:

- (i) compare a query index with an index of a field point representation for a record in the database;
- 10 (ii) identify the record as a hit if the comparison satisfies a search criterion; and
- (iii) repeat (i) and (ii) for a plurality of records.

Viewed from another aspect the present invention provides a graphical user  
15 interface to enable a user to interface with the database, comprising:

- an interface to enable a user to input data to the database; and/or
- an interface to enable a user to output data from the database; and/or
- an interface to enable a user to delete data from the database; and/or
- an interface to enable a user to update data in the database; and/or
- 20 an interface to enable a user to browse the database; and/or
- an interface to enable a user to search the database; and/or
- an interface to enable a user to displaying search results.

The graphical user interface may be provided on the computer system defined  
25 above or on another computer system such as a client computer system.

Viewed from another aspect the present invention provides a database for implementation on a computer system, the database configured to support a plurality of records, each record comprising a field point representation representing field  
30 extrema for a conformation of a chemical structure.

In another aspect the present invention provides computer software configured to provide the database defined herein and in a further aspect provides a carrier medium carrying the computer software.

5       Viewed from yet another aspect the present invention provides a method of generating an index of a field point representation representing field extrema for a conformation of a chemical structure, wherein the index is a string with n elements, the method comprising:

- 10       (i)     generating a numeric identifier from a characteristic of the field point representation;
- (ii)     generating one or more numbers in a range from 1 to n in dependence on the numeric identifier;
- (iii)    incrementing the string elements that correspond to the one or more numbers; and
- 15       (iv)    optionally repeating (i) to (iii) for another characteristic of the field point representation.

In the case that the string is a bit string, the incrementing step will be one of setting the bit to 1 (or the reverse in the case that the bit string is initialised to ones rather than zeroes). On the other hand, when the string elements are many-valued bins, 20 the bin value is incremented until its maximum is reached.

The method may further comprise using a deterministic function to generate one or more numbers in a range from 1 to n in dependence on the numeric identifier. 25

Viewed from yet another aspect the present invention provides a method of searching a database having a plurality of records, each record comprising a field point representation representing field extrema for a conformation of a chemical structure and having an index of the field point representation, the method 30 comprising:

(i) comparing a query index with an index of a field point representation for a record in the database;

(ii) identifying the record as a hit if the comparison satisfies a search criterion;

5 (iii) repeating (i) and (ii) for a plurality of records; and

(iv) outputting a representation of the records identified as a hit.

**BRIEF DESCRIPTION OF THE DRAWINGS**

For a better understanding of the invention and to show how the same may be carried into effect reference is now made by way of example to the accompanying  
5 drawings in which:

Figure 1 is a flow diagram illustrating the steps in the generation of a fieldprint;

10 Figure 2 is a flow diagram illustrating the steps performed for fieldprint searching;

Figures 3-14 illustrate interfaces of a graphical user interface for accessing the database;

15

Figure 15 is a flow diagram illustrating an import process;

Figures 16-19 illustrate further interfaces of the graphical user interface;

20 Figure 20 is an overview of the database;

Figure 21 illustrates the database schema; and

Figure 22 is a schematic representation of a computer system.

## DETAILED DESCRIPTION

The present invention relates to a computer system comprising a database having a plurality of records, wherein each record comprises a field point representation representing field extrema for a conformation of a chemical structure.

The computer system comprises an indexing mechanism for generating a searchable index in the form of a bit string for each field point representation. A bit string is stored in the database for each record.

The computer system also comprises a searching mechanism for searching through the indexes stored in the database to identify field point representations that match the field point representation of a search query. Known searching algorithms can be used.

A graphical user interface (GUI) is provided to enable a user to interface with the database. A user can use the GUI to input data to and output data from the database, to search the database and to browse the database.

The following sections describe in more detail the field point representations, the generation of indexes, searching the database and the graphical user interface. After these sections an overview of a particular embodiment of a database is given, followed by a detailed description of the database structure of the particular embodiment and a description of a computer system.

### I. Field Point Representations

It is possible to predict the binding properties of a candidate molecule, or other chemical structure, by representing the physical properties of a molecule which are important in its binding to other molecules, and then assessing the similarity between

two such sets of physical properties, one for the candidate molecule and one for a well characterised molecule.

Accurate molecular modelling is possible using advanced quantum mechanics.  
5 However, the computational effort needed for quantum mechanics is prohibitive for most biologically relevant molecules.

An alternative approach is called molecular mechanics. The most common way of implementing molecular mechanics in three dimensions is to calculate and  
10 compare fields around a molecule, such as the steric (van der Waals) and electrostatic (Coulombic) fields. The principles of molecular mechanics are simple and empirical. Moreover, molecular mechanics is computationally fast enough to cope with large proteins and other biopolymers associated with drug design.

15 In molecular mechanics electrostatic properties of a molecule are defined by placing a point charge at the centre of each atom (atom-centred charges or ACCs). Many different methods for calculating or estimating the value of such point charges are described in the literature. The aim of ACC methods is to distribute the point charges in such a way that the resulting electrostatic field is as similar as possible to  
20 the true electrostatic field (as determined by quantum mechanics methods). The electrostatic field as approximated by ACCs is usually quite accurate at a distance from the molecule ( $>5\text{\AA}$ ), but can be quite inaccurate at the molecular surface.

To improve the quality of molecular mechanics models at the molecular  
25 surface, extended electron distributions (XEDs) have been developed. The XED method involves replacing the point charge at the centre of some atoms with a set of point charges, one at the centre of the atom and one or more others distributed around that atom a short distance away. The XED method is described in Vinter (1994) [5] and Vinter and Trollope (1995) [6]. In the XED method, the XEDs themselves are  
30 treated simply as extra atoms which have charge but no volume. XED methods can

therefore calculate electrostatic interactions more accurately than ACC methods, while retaining the speed advantages of the molecular mechanics framework.

Quantum mechanical models and molecular mechanical models, such as ACC  
5 or XED models, can use the concept of field points to represent the molecular field. In  
this approach, the conformation of a molecule, i.e. its equilibrium arrangement either  
in isolation or when bound to another specific molecule or surface, is represented by a  
set of field points which measure field strength at a relatively small number of field  
maxima and minima around the molecule which are relevant to how the molecule is  
10 likely to interact with other molecules.

In order to calculate field points, a field definition must be adopted. One  
known field definition for molecular mechanical models uses positive and negative  
electrostatic interaction fields in combination with a surface interaction field. The two  
15 electrostatic interaction fields are defined by the interaction energy of a specific  
charged 'probe' molecule with the molecule of interest. For example, a probe the size  
of an oxygen atom, with either a +1 or a -1 unit charge, can be used. The field value  
at a given point is the interaction energy of the molecule with the probe atom sited  
with its centre at that point. The surface interaction field is defined by the van der  
20 Waals interaction energy of a neutral 'probe' with the molecule, for example an  
uncharged oxygen atom.

Other field definitions have been used, for example ones that include  
electrostatic fields calculated from quantum molecular methods, and ones that include  
25 hydrophobic fields calculated from the electrostatic field and its partial derivatives. In  
principle, any field definition can be used provided that its value can be defined at any  
point in space around the molecule.

Once the field definition has been made, the field points of the molecule need  
30 to be calculated. With the molecular modelling approach, the field points are

subdivided into a number of subsets, one for each field type, with each subset being calculated separately. The field points for a molecule are the values and locations of the extrema of its field, i.e. local maxima and minima. The final set of field points from each field type can be filtered to remove duplicate extrema and extrema with  
5 small energy values if desired.

The field point set encodes a large amount of information about the properties of the molecule, especially regarding its interaction with other molecules. The electrostatic field points encode information about the preferred hydrogen-bonding  
10 environment of the molecule, while the surface interaction field points encode the molecule's steric bulk.

The basic assumption underlying the field point approach is that two molecules which have similar sets of field points should have similar interactions with  
15 other molecules and hence should have similar biological activities. In other words, if molecule A has a certain biological activity, and molecule B is calculated to be similar to molecule A in a relevant conformation, then it is concluded that molecule B potentially has the same biological activity.

20 A field point representation therefore represents field extrema for a conformation of a chemical structure. Typically a field point representation includes a set of field points where each field point has a position and a field size value.

A field point representation may represent field extrema for a plurality of  
25 fields. In the example used herein the field point representation represents four fields, namely positive and negative electrostatic interaction fields, a surface interaction (i.e. steric) field, and a scaffold field.

Field point representations can be compared directly. For example, the  
30 similarity between conformations of two molecules can be calculated according to a



scoring formula which is sensitive to differences between the field point positions and energy values of the field points in the two field point sets.

5 However, it is desirable to generate a searchable index of a field point representation so that indexes can be stored in the database and searched upon to perform a screen out before further comparisons of the search results are performed, if required. Generating searchable indexes of a field point representation is non-trivial.

10 Field point representations are also referred to as field patterns herein and the terms can be used interchangeably.

## II. Index Generation

15 A searchable index of the field point representation is created in the form of a fingerprint-type bit string.

A fingerprint is generated from the molecule using a fingerprinting algorithm that examines the molecule and generates a pattern. Typical examples that are used include a pattern for each atom; a pattern for each atom and its nearest neighbour plus  
20 the joining bond; a pattern for each atom, its nearest neighbour, joining bond and further neighbours and bonds for varying path lengths; and a pattern for augmented atoms. The list of patterns produced is exhaustive, such that every pattern in the molecule up to the specified path length limit is generated. Each pattern serves as a seed to a pseudo-random number generator (i.e. it is hashed). The output of the  
25 pseudo-random number generator is a set of bits (typically 4 or 5 per pattern) which is added to the fingerprint with a logical OR. The creation of the seed is coded so as to produce a unique value for the pattern and hence the random number generation. Because each set of bits is produced by a pseudo-random number generator, it is likely that some bits will overlap. However, by setting 4 or 5 bits per pattern the probability  
30 that keys will be identical is reduced to an insignificant level for screen out purposes.

The size of the bit string may be set independently since, unlike keys, a bit does not have an exact meaning in the fingerprint. A bit string size of 2K (2048 bits) is commonly used as a compromise between speed and overlap. However other fingerprint sizes such as 1K, 4K and 8K could be used.

5

Fingerprints have the important property that, if a pattern is a substructure of a molecule, every bit in the pattern's bit string will be set in the molecules bit string. This means that simple boolean or bit-wise operations can be used. Each bit of a fingerprint can be thought of as being shared among an unknown but large number of patterns. Each pattern generates its particular set of bits. So long as at least one of those bits is unique, it can be established if the pattern is present or not. If a fingerprint indicates a pattern is missing then it certainly is, but it can only indicate a patterns presence with some probability. Since fingerprints have no predefined set of patterns, one fingerprinting system can be used to serve all databases and all types of queries.

15

Although not used in the current implementation, the fingerprint may be folded. Folding is a term used to describe a process whereby a fingerprint is halved in size by performing a logical OR on each half of the fingerprint. The result is a shorter fingerprint with a higher bit density. One can continue to fold until the desired bit density is achieved. With each fold one increases the chances of a false positive but one saves half the space required to store the fingerprint. Since one can only compare fingerprints of the same length some work must be done when querying to ensure there are bit strings of suitable length available for comparison.

20

25

Bit string theory is described in Mooers (1951 and 1956) [3, 4]. The basic principles that can be used and some advanced techniques which may be applied to bit strings will now be described.

Bit strings are an array of bits that are either set to zero or one (True or False). The length of the bit strings can vary depending on the type of index being created.

When the presence of a substructure is tested the bit strings are compared using a logical AND. For example, consider the following two 8 bit bit strings A and B.

A: 10100100

B: 11100110

10

Imagine both have been created using the same indexing method for the characterisation of a molecule B and a substructure query A.

One can test to see if the substructure is likely to exist in the main molecule by testing the following equation as true or false

$B \& A = A$  where  $\&$  is logical AND

For the example above a true result is produced, however if A is replaced with 10010100 a false result is produced. So one would know for certain that the substructure does not exist and should not waste time analysing the molecule further.

An exact match can be tested for by using  $B \& A = B$

The present system implementation allows bit strings to be compared for similarity using Tanimoto coefficient, Euclidian distance or Tversky similarity comparison techniques, each of which is now briefly described. Other bit-string comparison algorithms could also be provided.

The Tanimoto coefficient can be described as the number of bits in common between two bit strings divided by the total number of bits. This is an intuitive similarity measure as it is normalised to account for the number of bits that might be in common relative to the number that are in common. The equation can only be used  
5 as a similarity metric.

For two bit strings A and B their Tanimoto similarity is given by the equation

$$TS = BCm / (BCa + Bcb) - BCm$$

10 where

BCm is the number of bits set to 1 in common between the two bit strings

BCa is the count of bits set to 1 in bit string A

BCb is the count of bits set to 1 in bit string B

15 The results from this comparison range between 0 and 1, with 0 being the least similar and 1 being the most similar.

Euclidian distance is a measure of the geometric distance between two fingerprints, where each is thought of as a vector in multi-dimensional space. It can  
20 be used as a measure of similarity and as a substructure search metric depending on how it is applied.

Tversky similarity provides a most powerful metric. Like the Tanimoto metric, Tversky compares the features in a query bit string to features in the given  
25 (database) bit string. However, Tversky allows one to specify the weighting that will be given to each set of features. This allows the Tversky metric to be used in similarity, substructure and superstructure searching. The basic weightings are usually between 0 and 1 (0-100%) giving a ratio model. However the equation can be modified to accept weightings >100% thus providing a contrast model which causes

distinguishing features to be emphasised more than the common features which may be more useful in diversity or dissimilarity metrics.

For two bit strings A and B there Tversky similarity is given by the equation

5 
$$TvS = BCm / (\alpha BCa + \beta BCb) - BCm$$

where

BCm is the number of bits set to 1 in common between the two bit strings

BCa is the count of bits set to 1 in bit string A

10 BCb is the count of bits set to 1 in bit string B

$\alpha$  is the weighting to be given to bit string A

$\beta$  is the weighting to be given to bit string B

15 If both weightings are set to 100 then the Tversky equation gives the same results as the Tanimoto similarity. By varying the weightings the user can adjust how the bit strings are compared in terms of sub or super pattern similarity between the two bit strings.

20 Instead of the fingerprint bit string indexes used in the current implementation, data dictionary bit string indexes could be used.

25 Data dictionary indexes are also known as structural keys. A structural key is represented as a boolean array in which each element is true or false. Boolean arrays in turn are represented as bit strings in which each bit represents one position of the boolean array. A structural key is a bit string in which each bit represents the presence (true) or absence (false) of a specific structural feature (pattern). A fragment library is created of the patterns that are considered important, each pattern being assigned to a bit of the bit string. The number of fragments in the library dictates the bit string length. The bit string for a molecule is created by carrying out a substructure search  
30 of each structure or pattern in the fragment library and setting its corresponding bit in

the bit string appropriately. Depending on the number of fragments in the library this can be a time consuming process. When a database is searched for a particular structural feature, a search key is generated. As the search proceeds, the search key is compared to the bit string of each molecule in the database. If a TRUE bit in the search key is not also set as TRUE in the molecule's key, then the structural feature represented by that bit is not in the molecule, so the molecule can be excluded from consideration.

Structural keys, like fingerprints, have the important property that, if a pattern is a substructure of a molecule, every bit in the pattern's bit string will be set in the molecule's bit string, thus allowing boolean or bit-wise operations to be used to compare bit strings.

Using bit strings as indexes allows rapid bitwise comparison using simple AND, OR, XOR and NOT computer operations. They are also particularly suitable to use in similarity measures based on the numerous similarity formulae that exist. The method by which data is encoded into a bit string is known as fingerprinting. Whilst the use of fingerprinting and bit strings is known, the approach has never been applied to field point representations. In other words generating bit strings from field point representations is new.

In one embodiment an indexing mechanism is used to generate an index of a field point representation. The indexing mechanism may be implemented on a computer system as software, firmware or hardware, although in a particular embodiment it is implemented as software.

In a particular embodiment the index is a bit string of length  $n$  and the indexing mechanism is configured to:

- (i) generate a numeric identifier from a characteristic of the field point representation;

- (ii) generate one or more numbers in a range from 1 to n in dependence on the numeric identifier;
- (iii) set the bits in the bit string that correspond to the one or more numbers; and
- (iv) optionally repeat (i) to (iii) for another characteristic of the field point representation.

Thus, starting with a bit string of length n with all n bits set to zero (or indeed with all n bits set to 1), bits of the bits string can be set in dependence on one or more characteristics of the field point representation. Suitably, one or more characteristics are identified, one or more numeric identifiers are generated, and one or more numbers between 1 and n are generated. These features will now be described.

#### II.A. Characteristics

The characteristic of the field point representation can be any property and/or relationship that exists within the data.

The properties that can exist in a field point representation include the field type of each field point (for example negative, positive, surface, scaffold); the size or energy of each field point; the total number of field points; the number of each type of field point; and the X, Y, Z coordinates of a field point.

Relationships which can be derived from the properties include the pairwise distance relationship between two field points; the angles between three field points; the triangulation distances between three field points; any other relationship of interest between two or more field points

Any or all of the properties and relationships may be used by the indexing mechanism or a fingerprinting algorithm to generate an index (fingerprint) from a given field point representation (field pattern).

In one embodiment a characteristic of the field point representation includes one or more of:

- the number of field points of a particular field of the field point representation;
- 5 the particular field and energy of a field point in the field point representation;
- and
- the respective energies of and distance between a field point pairing in the field point representation.

10 A characteristic of the field point representation is used to generate a numeric identifier which in turn is used to generate one or more numbers between 1 and n for setting bits in the bit string. In order to understand the generation of the numeric identifier from a field point representation, the generation of one or more numbers between 1 and n in dependence on the numeric identifier will first be described.

15

#### II.B. Generation of Numbers between 1 and n

In one embodiment the indexing mechanism is configured to generate one or more numbers in a range from 1 to n in dependence on the numeric identifier by using  
20 a deterministic function.

A deterministic function is a function which takes a value as an input value or seed and generates one or more output values in dependence on the input value such that the one or more output values for any given input value is always the same.

25

For example, if a deterministic function is seeded with the number 27 to produce four output values, it may output the values 0.23, 0.33, 0.21 and 0.88. If the same function is subsequently seeded with the number 27, then it will output the same four values, namely 0.23, 0.33, 0.21 and 0.88.

30



Deterministic functions can be used to generate one or more integer output values between 1 and a number n, by converting the output values to integers in this range. This can be done by scaling and rounding the output values.

5        For example, certain deterministic functions can generate all output values between 0 and 1. These can be scaled to an integer value between 1 and n by using the formula:

$$\text{integer value} = \text{ROUND}(\text{output value} * (n-1) + 1)$$

10

An integer value generated in this way can be used to set a corresponding bit in a bit string. If, for example, the deterministic function is seeded to produce four output values from one seed (input value) then four integer values can be generated and used to set four bits in the bit string.

15

Examples of deterministic functions are hashing algorithms and pseudo random number generators. The current system implementation uses a pseudo random number generator.

20        In one embodiment known length bit strings are used. Starting with a bit string containing only a series of 0's, the basis of the approach is to create a unique identifier (number) for each and every property or relationship contained within the field pattern. The unique identifier is used as a seed to initialise a random number generator. The random number generator is used to provide a series a numbers  
25 (commonly 4 numbers) between 1 and the length of the bit string. The numbers produced are used to set the corresponding bit in the bit string to 1. After cycling around all the properties or relationships that are to be analysed, the bit string will contain a series of 0's and 1's which are unique to that field pattern.

An important part of creating any bit string index is to create the unique identifier for a defined property or relationship. Once created, the unique identifier will always produce the same sequence from a deterministic function.

5     II.C. Generation of the Numeric Identifier

The indexing mechanism can be configured to take a measurement of a characteristic to generate the numeric identifier.

10         In a particular example for generating a bit string (including the generation of the numbers in a range from 1 to n), the indexing mechanism uses the fingerprinting algorithm detailed below in pseudo code. The code is applied to each field point representation (field pattern) being stored in the database giving an index (fingerprint) for each record.

15         The code is exemplified using a bit string length of 2048 however; bit strings of any appropriate length can be used.

1. A bit string of length 2048 is created consisting entirely of 0's (zeros)
- 20    2. For each field type (negative, positive, surface, scaffold)
  - a. Count the number of field points of that type in the pattern.
  - b. Encode the field type and the field point count into a preferably unique numeric identifier
  - c. Seed a pseudo random number generator with the numeric identifier
  - 25    d. Obtain four numbers from the pseudo random number generator between 0 and 2047 (to span a range from 1 to 2048 and use them to set the corresponding bit in the bit string to 1.
3. For each field point in the pattern
  - a. Encode the field type and the field point energy into a preferably unique
  - 30    numeric identifier

- b. Seed a pseudo random number generator with the numeric identifier
- c. Obtain four numbers from the pseudo random number generator between 0 and 2047 and use them to set the corresponding bit in the bit string to 1.
- 4. For each field point pairing in the field pattern
- 5 a. Calculate the distance (to a given precision) between the two points from their X, Y, Z coordinates.
- b. Encode the two field types and distance between them into a unique numeric identifier
- c. Seed a pseudo random number generator with the numeric identifier
- 10 d. Obtain four numbers from the pseudo random number generator between 0 and 2047 and use them to set the corresponding bit in the bit string to 1.

Figure 1 illustrates a fingerprint generation method. It is noted that the flow diagram refers to bins rather than bits. However, the bins in this embodiment can only adopt values of 0 or 1, so that bin and bit are synonymous. In the more general case where each bin can adopt an arbitrary number of values, the step of "Set all bins to 0" will be the same, but the step of "Set corresponding bins to 1" will become one of incrementing the bin values.

20 The resulting fingerprint bit string contains a series of 1's and 0's which encodes the nature of the field pattern. The fingerprint generated is then stored in the database.

25 In step 4 it is possible to alter the precision at which the distance between two field points is measured. In the current example four precision levels (1, 0.5, 0.25 and 0.1 Angstroms) are used.

This means that for each field pattern registered to the database four Fingerprints are generated and stored in the database. This allows searches to be carried out over the database at different precision levels. Thus it will be appreciated

30

that in one embodiment the indexing mechanism is configured to take a measurement of a characteristic at different levels of precision to generate corresponding multiple indexes which represent the field point representation at different precision levels.

5           Other methods can be used to encode the field pattern into a bit string. For instance three field point comparisons (Triangles) may be used rather than the two field point comparison detailed above. In this case the same procedures as outlined above can be used except in section 4 the information for each three field point grouping would be encoded.

10

In another embodiment the indexing mechanism is configured to:

- (i)     define a plurality of ranges of possible measurement values;
- (ii)    take a measurement of a characteristic of the field point representation to produce a measurement value;
- 15   (iii)   assign the measurement value to a range if the measurement value is within the range;
- (iv)    optionally repeat (ii) and (iii); and
- (v)     use the number of measurement values assigned to a range to generate the numeric identifier.

20

In a particular example which uses a definition of a plurality of ranges, a numeric identifier is generated for each field point pair and used as a 'seed' for a pseudo-random number generator. Measurements are taken of the following characteristics:

- 25           - the field type (one of four) for each field point  
            - the field energy for each field point  
            - the distance between the field points

There are 10 possibilities since there are 10 possible combinations of 4 field  
30 types, and these can therefore be encoded into a number between 1 and 10.

Ranges with a width that can be considered as an 'energy precision parameter' are defined for the energies. These ranges are used to convert each field point energy (measurement value) into an integer. For example:

- 5                    0-5 becomes 1
- 5-10 becomes 2
- 10-15 becomes 3

and so on.

- 10                  The energy precision parameter determines the width of the ranges, which in the example above is 5.0. This means that field points with energy values between 0 and 5 are considered to be the 'same', those between 5 and 10 are the 'same' and so on.

- 15                  The field point pair distance needs to be similarly encoded. Suitably, each possible distance is assigned an integer, such that if two distances are to be considered the 'same' then the integer assigned to them should be the same.

One method uses a constant distance resolution or precision level, so:

- 0 - 1 becomes 1
- 20                  1 - 2 becomes 2
- 2 - 3 becomes 3

and so on. This example has a distance resolution of 1, as all distances are rounded up to the nearest 1 Angstrom.

- 25                  One example uses 4 'precision levels' which correspond to different distance resolutions. In the example the 4 distance resolutions are 0.25, 0.5, 1.0 and 2.0. At 0.25, for example, the mapping is such that:

- 0 - 0.25 becomes 1
- 0.25 - 0.5 becomes 2
- 30                  0.5 - 0.75 becomes 3

and so forth.

In another example a lookup table is used to define the ranges and map the distances to integers. This removes the constraint that the distance resolution needs to be the same at all distances. For example, higher resolutions can be used at short distances, while lower resolutions can be used at long distances. In an example the mapping is such that:

	0 - 0.1 becomes 1
10	0.1 - 0.2 becomes 2
	0.2 - 0.4 becomes 3
	0.4 - 0.7 becomes 4
	0.7 - 1.0 becomes 5
	1.0 - 2.0 becomes 6
15	2.0 - 5.0 becomes 7
	5.0 - 10.0 becomes 8
	10.0 - 20.0 becomes 9
	> 20.0 becomes 10

Thus in this example any distance is mapped to a number from 1 to 10 and distances of 0.23 and 0.53 are seen as 'different', but distances of 11.0 and 17.0 are the 'same', for example.

Once four integers for the field point pair have been generated (the one representing field types, the two representing the field sizes, and the one representing the field distance), these can be combined into a single integer for the field point pair.

For example, if the field types integer can be 1-10, the size values can be 1-10, and the distance value can be 1-100, then

$$K = (\text{distance value}) * 1000 + (\text{size value 1}) * 100 + (\text{size value 2}) * 10 + (\text{types value})$$

5 encodes these four numbers into one number K in such a way that each value of K uniquely maps to a (dist, size1, size2, types) set. This number K is the numeric identifier which is then used as the seed to the hash function or pseudo random number generator which is used to set one or more bits in the bit string.

10 Thus it will be appreciated that using the above the indexing mechanism can be configured to define ranges of equal width across all ranges or to define a range for smaller measurement values with a narrower width than a range for larger measurement values. In a particular embodiment the indexing mechanism is configured to generate multiple indexes by defining ranges of different widths for different precision levels.

15 Indexes in the form of bit strings representing field point representations are stored in a database to allow rapid searching of field point representations. The following section describes some techniques used to compare a search query with indexes in the database.

20

### III. Searching the Database

25 Since a known index in the form of a bit string is used in particular embodiments of the present invention, known bit string manipulation techniques can be used, such as testing for substructures, testing for exact matches, Tanimoto coefficient testing, Euclidian distance testing and Tversky testing.

In one embodiment a searching mechanism is used to search the database. The searching mechanism may be implemented on a computer system as software,

firmware or hardware, although in a particular embodiment it is implemented as software.

Suitably, the searching mechanism is configured to:

- 5 (i) compare a query index with an index of a field point representation for a record in the database;
- (ii) identify the record as a hit if the comparison satisfies a search criterion; and
- (iii) repeat (i) and (ii) for a plurality of records.

10 The plurality of records can be all of the records in the database or a subset of these.

The searching mechanism can be further configured to:

- receive a search query identifying a field point representation; and
- 15 form the query index by generating an index of the field point representation identified by the search query.

In one embodiment the searching mechanism is configured to form the query index by using the indexing mechanism to generate an index of the field point representation identified by the search query. Suitably, the searching mechanism is  
20 configured to generate the query index as a bit string.

The processes involved in the execution of field pattern searching in a particular example are given below.

25

1. Using a GUI a user selects
  - a. The field pattern to be used as the query. This may be from:
    - i. A conformation field pattern already registered to the database.
    - ii. An external file in the XED format (the system could be  
30 developed to allow external files in other formats to be used)



- b. The comparison type to be used for the search.
- c. If a similarity comparison is chosen the user is required to provide the maximum and minimum similarity range that will be regarded as a hit during the comparison.
- 5 d. The precision level at which the search should be carried out.
2. On submitting the query the GUI passes the information to the database.
3. The database then
  - a. Creates a fingerprint (bit string representation of the field pattern) for
  - 10 the query at the required precision level.
  - b. Creates a temporary table to hold the results.
  - c. Searches all of the fingerprint indexes (at the requested precision level) stored in the database.
  - d. Writes information to the temporary results table regarding any hit.
  - 15 e. When the search is complete the database informs the GUI in which table the results are held.
  - f. The GUI then selects the information from the table and displays it to the user.
  - g. Once the user has finished viewing the results the GUI tells the
  - 20 database to delete the table holding the results.

Figure 2 is a flow diagram illustrating the fingerprint searching for the particular example.

- 25 In a particular embodiment the searching mechanism is configured to use a true/false matching technique to compare a search query with a record. True/false matching techniques that can be used in the current embodiment include an exact pattern technique, a sub pattern technique and a super pattern technique.

The searching mechanism can also be configured to use a similarity measuring technique to compare the search query with the record. In one embodiment, similarity measuring techniques that can be used include a Euclidian distance technique, a streetcar distance technique, a sub pattern similarity technique, a super pattern  
5 similarity technique, a Tanimoto similarity technique, a dice technique, and a Tversky similarity technique.

The searching mechanism is configured to identify a record as a hit dependent on a similarity measure produced by the similarity measuring technique being in a  
10 range from a minimum similarity value to a maximum similarity value.

In a particular embodiment the searching mechanism is configured to search by precision level. Suitably, this is done by generating an index of the field point representation at a required precision level to form the query index and comparing the  
15 query index with an index at the same precision level of a field point representation for a record in the database.

A user can submit a search query through the graphical user interface (GUI). The searching mechanism stores the hits in a results table which is used to display the  
20 results to the user through the GUI. This and other functionality of the GUI will now be described.

#### IV. Graphical User Interface (GUI)

25 A GUI is provided to enable a user to interact with the database without having to use command line arguments in SQL. (The database can be used without a GUI by a suitably expert user.) The GUI allows a user to carry out complex operations by simple menu-driven button clicking.

The GUI of the current implementation is written to run in a Windows environment using Visual Basic. The GUI uses activex data objects (ADOs) to communicate with the database, however any other open database connectivity (ODBC) compliant connection protocols could be used.

5

The GUI is provided by a user application which has very little knowledge of the complexity of the internal database structure. The user application does however know what functions and procedures are available to it. The user application's main role is to ensure the correct calls are made to the database and any returned  
10 information is dealt with appropriately.

When the user application is started, the GUI displays the empty navigation form shown in Figure 3. By clicking the New Connection label in the navigation form a login dialogue is displayed as shown in Figure 4.

15

If the user has previously saved any login information it will automatically be displayed in the dialogue. The user must supply the information or check that the information is correct. The user may select Save Information box if he or she would like the login information saved for later use. Clicking the OK button establishes a  
20 connection to the desired database as long as the information is correct. By providing a login dialogue the user may connect to any Oracle database that may be provided.

Once connected to a database the application establishes what data are present and displays the name of the database and a series of navigation options as a tree in  
25 the navigation screen as shown in Figure 5.

By clicking the BROWSE branch, the tree is further extended to show two sub-branches, namely RECORDS and CONFORMERS as shown in Figure 6. By selecting RECORDS the tree further expands to show how database records are  
30 classified by their source in the database as shown in Figure 7.

In the example of Figure 7 only MAYBRIDGE and THROMBIN sources have associated database records. A source type of ALL is supplied by default allowing the browsing of all records within the database. By further clicking on a particular source  
5 the records for that source are further categorised by the records registration type as shown in Figure 8. In this instance only records with a type of MOLECULES exist within the selected MAYBRIDGE records.

By selecting either a particular source or type within the tree a record browsing  
10 form is displayed, allowing the user to view and navigate the desired records.

In the above dialogue if the label MOLECULES under the source MAYBRIDGE is selected the user will be shown a records browsing dialogue allowing he or she to view only the records classified as source MAYBRIDGE and of  
15 type MOLECULES.

Figure 9 shows the record browsing form. The form displays the data stored in the database for each record. The first structure conformation stored for the current record is used to display as a sample structure for the current molecule.  
20

Navigation buttons are provided allowing the user to scroll through the records in the data set or go directly to a particular record in the database. The records browsing form has two further menus. The "Admin" dropdown menu contains a "Record Delete" option which if selected removes the current record and all of its  
25 conformers from the database. The "Display" dropdown menu has options for the structure display and allows desired field points to be displayed with or without the structure. For example the same record with atoms and fields turned on can be displayed as shown in Figure 10.

A "View All Confs" button is provided on the record browsing form. This allows the user to view all of the conformers for that molecule held in the database.

5 The conformers form can be accessed by clicking the BROWSE→  
RECORDS→ CONFORMERS tree label (see e.g. Figure 6) in the database  
navigation form or by clicking the "View All Confs" button for a particular record.  
Figure 11 shows the conformers form. This form allows the user to scroll through all  
of the conformers in the database or the conformers for a particular molecule  
depending on how the form has been accessed.

10

The form has three menus. An "Admin" dropdown menu allows the user to  
delete the currently displayed conformation from the database. A "Display" dropdown  
menu is identical to that in the record browsing form. An "Options" dropdown menu  
allows the currently displayed conformations field pattern to be used as a query to  
15 search the database.

20

By clicking the ADMIN tree label within the database navigation form several  
administrations options are displayed. These allow the browsing and creation of  
source and types within the database as shown in Figure 12.

25

The DICTIONARY branch allows the user to add source and type  
information to the database. Types and sources are assigned to molecules on import.  
By clicking either the types or sources label a window allowing browsing or creation  
is displayed as shown in Figure 13.

The options under the CHEMISTRY label include IMPORT and EXPORT  
sub-branches which are to allow a user to initiate import or export of information to  
the database. The IMPORT screen is shown in Figure 14. The system can import  
files in a XED format and assumes that a single molecule and its conformations exist

in a single file. This is merely a convenient choice for the present implementation and the system could be readily adapted to allow import of different formats.

To import a file, such as one in the XED format, the user is required to supply certain information, including: whether all the files or a single file in the specified directory should be imported; the source for the molecules to be imported against (selected from a drop down list of items stored in the database); the type for the molecules to be imported against (selected from a drop down list of items stored in the database); and the directory that the files are located in.

Once complete the user may start the import process. This application checks the import directory for files and sends a command to the database to import each file in turn. After a file has been imported the file is deleted from the directory.

In a particular example, the following operations are performed for the import process:

1. GUI sends a call to the database to import a file supplying the filename, type and source information.

2. The database opens the file and reads the following information.

- a. The import file name
- b. The name of the molecule
- c. The description of the molecule

3. The database then calculates the following information based on a conformation in the file

- a. An ID for the molecule
- b. The molecular weight
- c. The molecular formula

4. The information gathered in 1, 2 and 3 are used to register an entry for this molecule in the object table.

5. The database then cycles through each conformation in the file and registers an entry for each in the Structures table using the following data:
  - a. A conformation sequence number (incremented by 1 for each conformation stored)
  - 5 b. The conformation energy (obtained from the file)
  - c. A representation of the structure and field pattern in binary format.
6. Once the conformations have been registered the database looks up the following information and updates the entry for the molecule in the object table
  - 10 a. The number of conformations for this molecule (i.e. the number of entries in the structures table for this molecule)
  - b. The maximum conformation energy
  - c. The minimum conformation energy
7. For each conformation in the file one or more fingerprints are generated from its field pattern. The fingerprints are then stored in the fieldprints table. The fingerprints are used in the field pattern searching techniques.

Figure 15 is a flow diagram illustrating a file import process.

- 20 The user may also select the continuous polling option. This provides a mechanism to allow file import to take place continuously. If selected, the application continuously monitors the import directory for new files and imports any its finds.

- 25 By selecting the EXPORT label in the navigation tree the user may export information from the database in a file format, such as the XED file format for example.

Figure 16 shows the export form. The user is required to supply a database record or a list of records (stored in a text file) in the database for export. The user

also supplies the filename to which the records will be written. The file produced contains an exact replica of the information originally used for the import process.

5 The indexes tree in the navigation form is used to test various aspects of the data in the database.

The options under the ANALYSE label within the navigation form allow a user to carry out a field pattern search as shown in Figure 17.

10 By selecting the FIELD SEARCH label a field search form is displayed from which a user is able to initiate a field pattern search across the database.

Figure 18 shows the field search form, the various parameter fields of which are now described.

15 The user supplies via the GUI a field pattern to be used as the query (see Select Data Source options). This may be from a database record, selected by use of the dropdown lists to use as the query field pattern or a file containing a molecule or molecules in a particular file format, for example a XED file format. Where multiple  
20 molecules exist per file the user may select the particular conformation to use.

25 The user selects the search type (see Select Search Type options). Where the search type is a similarity measure, the user can supply a range of allowed similarities values for the results. This is because all entries will have a value for similarity, so it is used to restrict the possible number of hits returned. The user can also supply query and target weighting values when the Tversky similarity comparison is used. These are particular to the Tversky equation and allow the user to tune the similarity calculation.



The user can select a required precision level for the search (see Select Search Index Type options). This dictates which fingerprint index will be used in the search. The precision levels of the particular embodiment are set at Level 1 = 1Å, 2 = 0.5Å, 3 = 0.25Å and 4 = 0.1Å. Other values may be used.

5

The user may select the criteria to be used in the query fingerprint generation (see Set Criteria options). Usually all of these options will be selected, but this may not always be desired. For example, the field definition may be restricted to omit the scaffold field (Sca), or to omit the negative and positive electrostatic (Neg, Pos) and surface interaction (Sur) fields.

10

When the search is initiated the application sends the information to the database. The database uses this information to calculate the fingerprint and compares it to all of the fingerprints stored in the database. The database writes the results to a temporary table and returns the name of the table containing the results to the application. The application then displays a results form with the information from the search.

15

Figure 19 shows a sample results form.

20

The results are initially ordered by descending similarity (if applicable). The results are displayed in a list showing the following information: the ID of the molecule; the similarity measure (if a similarity search has been performed); the source of the molecule; and the type of the molecule

25

The "Display" dropdown menu allows the display options for the query structure and the currently selected hit structure to be altered. As can be seen from the figure, these are displayed to the right side of the list. The user is able to select an entry in the displayed results by clicking on a row in the list. The form is updated to show the structure of the currently selected hit.

30

The "Data Items" dropdown menu allows a user to change the information shown in the list. The choices given to the user are: ID; Conformation number;  
5 Similarity; Source; Type; Conformer Energy; Molecular Weight; Name; Description; and Import Filename.

The "Options" dropdown menu allows the user to view the Information of the currently selected hit in a Browse Record window; view the conformations of the  
10 currently selected hit in a Browse Conformers window; use the currently selected hit as a query in a new search.

The "File" dropdown menu allows the export of a comma separated text file containing the information displayed in the form; a list containing only ID and  
15 Conformer information

When the "Results" window is closed the application instructs the database to delete the temporary table holding the information.

## 20 V. Database Overview

Figure 20 shows an overview of the database. In the illustrated embodiment the database 100 is as an Oracle database (version 8.1.7 or greater). A separate user application 102 provides the GUI which is configured to enable a user to interface  
25 with data stored in the database. Files 104 containing structure data, including data representing field point representations, are also illustrated.

Import operations (illustrated as 1 in Figure 20) include importing data from the files 104 to the user application 102, transferring data from the user application  
30 102 to the database 100 and transferring data from the files 104 directly to the

database 100. Export operations (illustrated as 2) include transferring data from the database 100 to the user application or to files 104. Searching (illustrated as 3) can be performed using the user application 102, optionally using data from a file 104. Browsing the database (illustrated as 4) can be performed using the GUI of user application 102.

The database comprises tables 106 comprising data 108 and views 110 for viewing data split across more than one table. The database also comprises packages 112 comprising public functions and procedures used by the user application and private functions and procedures used internally to execute particular tasks (for example to execute searching). The database also comprises sequences 114 for providing consecutive numbering for items in the database.

Referring back to the index mechanism and the searching mechanism, these are implemented as software functions/procedures in the database of the illustrated embodiment.

Creation and maintenance of the features within the database are achieved using conventional techniques and methods supported by the Oracle database environment. In the illustrated embodiment all procedures and functions have an SQL interface and the code executed by the procedure or function may be implemented in SQL or Java.

It will be appreciated that in the illustrated embodiment much of the functionality of the system is embedded within the database itself, for example for storing data, retrieving data and searching data. Communication between the GUI/user application 102 and database 100 is achieved using conventional protocols, for example ADO although any suitable protocol can be used.

The user application 102 is written in Visual Basic and may be run in any standard Windows PC environment. In the most part the GUI communicates with the database through the packages embedded within the database. The GUI can also directly access data from the tables for display purposes, such as record browsing.

5

The GUI enables a user to input data to the database, to output data from the database, to delete data from the database, to update data in the database, to browse the database, to search the database, and to display search results.

## 10 VI. Database Structure

This section details the physical structure of the database schema of a particular embodiment. An overview of the tables of the database schema is given in Figure 21.

15

The database schema is centred on the Objects table. This holds the top-level Information for each molecule registered. Each Molecule has a single entry in the objects table and is uniquely identified by a specific ID allocated at registration. This ID is used throughout the other tables in the schema to identify items related to that molecule. The structures table holds all of the structure information (an entry per conformation) for each molecule. This allows the structure of any conformation to be retrieved, interpreted and displayed by a suitable application connecting to the database. In the particular embodiment the structure information is held within the table as a Binary Large Object (BLOB) data-type.

20  
25

General properties for each molecule are held in the objects table, whilst properties specific to a conformation are held each in the structures table.

When a molecule is registered to the database a Type and Source must be supplied. These must match allowed items for the Type and Source defined in the Type\_Dict and Source\_Dict tables.

5           The Source identifier allows the association of a molecule and hence its conformations with a particular source. The user may give any name to a source that has meaning to them. This could be used to track companies or projects within the database, for example MDR, HIV, or MayBridge.

10           The Type identifier allows the association of a molecule and hence its conformations with a particular type. The user may give any name to a type that has meaning to them. This could be used to track different entity types, for example Molecule, Fragment, Building Block or Field Template.

15           Any number of source and types can be created in the database, however only one source and type can be associated with a given molecule and its conformations.

20           The chemical structures stored in the Structures table are a complete representation of the information supplied at registration time i.e. chemical structure and field point representation (field pattern). However they are not used for searching. The schema provides a separate Fieldprints table to hold data generated at registration time which is more applicable to field searching.

#### VI.1 Tables

25

The tables of the schema will be described in turn.

#### OBJECTS Table

The objects table holds the top-level information for each entry in the database. One entry per molecule will exist in this table.

Where data integrity is to be maintained constraints have been created, i.e. it is not possible to register an entry to the table with an ID that already exists, or with a TypeID or SourceID that does not exist in the appropriate table.

Table Structure

FIELD	DATA TYPE	NUL L	DESCRIPTION	Constrai nt	Constraint LINK
OBJECTID	NUMBER (11)	N	Internal ID created from a sequence	PKEY, UNIQUE	
NAME	VARCHAR2 (255)	N	Supplied data from import file		
DESCRIPTION	VARCHAR2 (255)	Y	Supplied data from import file		
TYPEID	NUMBER (11)	N	Supplied data from list of allowed types	FKEY	Type_Dict:Typeid
SOURCEID	NUMBER (11)	Y	Supplied data from list of allowed dictionary sources	FKEY	Source_Dict:Sourceid
MOLFORMULA	VARCHAR(255)	Y	Calculated from the structure		
MOLWIEGHT	NUMBER (11,4)	Y	Calculated from the structure		
NUMSTRUCTURES	NUMBER (11)	Y	Calculated from the number of entries for this molecule registered in structures table.		
MAXENERGY	NUMBER 11,4	Y	Calculated from the max energy of the conformations registered for this molecule in the structures table		
MINENERGY	NUMBER 11,4	Y	Calculated from the min energy of the conformations registered for this molecule in the structures table		
IMPORTFILE	VARCHAR2 (255)	Y	The file the molecule was imported from		
TIMESTAMP	NUMBER 15	N	System assigned registration date		

### STRUCTURES Table

The structures table holds data about each and every conformation loaded into the database. A sequence number is assigned internally to differentiate the conformers for a particular molecule.

5

Table Structure

FIELD	Data TYPE	NUL L	DESCRIPTION	Constrai nt	Constraint LINK
OBJECTID	NUMBER (11)	N		FKEY	Objects:Objectid
STRUCTURESEQ NO	NUMBER (11)	N	The particular number of conformation stored for this molecule		
STRUCTURE	BLOB	N	Binary storage of the structure from the import file		
ENERGY	NUMBER 11	Y	Supplied data from import file		
TIMESTAMP	NUMBER 15	N	System assigned registration date		

### FIELDPRINTS Table

10 The Fieldprints table holds the data created for searching of the field point representation or field pattern. In the particular embodiment this data is created at various precision levels. Each precision level has an entry within the table. In the particular embodiment four precision levels are used.

15 A fingerprint is created for each and every conformation stored in the database from its field point representation. All fingerprints of the same precision level are combined into a single blob for rapid searching.

Table Structure

FIELD	Data TYPE	NUL L	DESCRIPTION	Constrai nt	Constrai nt LINK
IDXLEVEL	NUMBER (11)	N	The precision level at which the index was created	PKEY	
IDXPRINT	BLOB	N	The blob containing data at specified precision for all structures containing fields		
TIMESTAMP	NUMBER 15	N	System assigned registration date		

### TYPE\_DICT Table

This table stores all of the dictionary items that may be assigned to the molecule being registered.

5

#### Table Structure

FIELD	Data TYPE	NUL L	DESCRIPTION	Constrai nt	Constrai nt LINK
TYPEID	NUMBER (11)	N	Internal ID created from a sequence	PKEY, UNIQUE	
NAME	VARCHAR2 (255)	N	User supplied data		
DESCRIPTION	VARCHAR2 (255)	Y	User supplied data		
TIMESTAMP	NUMBER 15	N	System assigned registration date		

### SOURCE\_DICT Table

10 This table stores all of the dictionary items that may be assigned to the molecule being registered.

#### Table Structure

FIELD	Data TYPE	NUL L	DESCRIPTION	Constrai nt	Constrai nt LINK
SOURCEID	NUMBER (11)	N	Internal ID created from a sequence	PKEY, UNIQUE	
NAME	VARCHAR2 (255)	N	User supplied data		
DESCRIPTION	VARCHAR2 (255)	Y	User supplied data		
TIMESTAMP	NUMBER 15	N	System assigned registration date		

15

### RESULTS (X) Table

This table stores the results obtained from any fingerprint search and is transitional.



Each fingerprint search will have its own results table created and is identified by the \_(X) part of the table name. The X is assigned internally as the next number from a sequence.

- 5 This table is usually deleted when no longer required by the user application

Table Structure

FIELD	Data TYPE	NUL L	DESCRIPTION	Constrai nt	Constrai nt LINK
OBJECTID	NUMBER (11)	N			
OBJECTIDSEQN O	NUMBER (11)	Y			
SIMILARITY	NUMBER (6)	Y			

10 VI.2 Views

The following views provide easier access to complete data listings stored across multiple tables.

15 OBJECTVIEW View

This view provides a look at data in the Objects table including all of the property values stored for those compounds.

FIELD	Table
OBJECTID	Objects
NAME	Objects
DESCRIPTION	Objects
TYPE	Type
SOURCE	Source
MOLFORMULA	Objects
MOLWIEGHT	Objects
NUMSTRUCTU RES	Objects
MAXENERGY	Objects
MINENERGY	Objects
IMPORTFILE	Objects
TIMESTAMP	Objects

## RESULTS (X) VIEW

The RESULTS\_(X)\_VIEW view is created when a RESULTS\_(X) table is created. This allows easier access to extra information about the molecules and conformations retrieved in a fingerprint search.

5

FIELD	Table
OBJECTID	Structures
OBJECTIDSEQN O	Structures
SIMILARITY	Results (x)
SOURCE	ObjectView
TYPE	ObjectView
ENERGY	ObjectView
MOLWEIGHT	ObjectView
NAME	ObjectView
DESCRIPTION	ObjectView
IMPORTFILE	ObjectView

## VI.3 Sequences

Sequences are used for the creation unique identifiers. The following sequences are  
 10 used in the current implementation.

An OBJECTID\_SEQ sequence is used to create the unique identifier for each entry in the OBJECTS table, i.e. the ObjectID data.

15 A TYPEID\_SEQ sequence is used to create the unique identifier for each entry in the TYPE\_DICT table, i.e. the TypeID data.

A SOURCEID\_SEQ sequence is used to create the unique identifier for each entry in the SOURCE\_DICT table, i.e. the SourceID data.

20

A RESULTS\_SEQ sequence is used to create the unique identifier for each results table name, i.e. the X part of the title.

#### VI. 4 Database Packages

The use of functions and procedures within the Oracle database environment allows complex tasks to be completed with a single call to the database. They also provide a way of masking the complexity of the database to a user or application, i.e. the user does not have to know the internal detail of the database schema, to register various bits of information, they need only supply the data to a procedure or function happy in the knowledge that the method knows how to deal with it.

Functions and procedures can also be amalgamated into packages. In the present implementation, the call interface for all functions and procedures is declared using SQL since this is the language of the database environment. However the executable code may be written in SQL, C, Java, or a mixture of these languages.

The use of packages allows procedures and functions to be specified as public and private. Calls made externally to the database may only use public methods.

The database environment of the present embodiment has three packages. One package (PACK\_CBMD\_REG) is concerned with registration of molecules and their conformations along with all of the information (such as the fingerprints) into the database tables. A second package (PACK\_CBMD\_CHEM) is concerned with searching the fingerprint (the indexes). A third package (PACK\_CBMD\_UTILS) contains general utilities used by the other two packages.

Annex A details the packages and their public interfaces including their public procedures and functions which may be called externally from the database. The languages used and a general description are also given.

#### VII. Computer System

Figure 22 shows a schematic and simplified representation of a computer system 200. The computer system 200 comprises various data processing resources such as a processor (CPU) 230 coupled to a bus structure 238. Also connected to the bus structure 238 are further data processing resources such as read only memory 232 and random access memory 234. A display adapter 236 connects a display device 218 having screen 220 to the bus structure 238. One or more user-input device adapters 240 connect the user-input devices, including the keyboard 222 and mouse 224 to the bus structure 238. An adapter 241 for the connection of the printer 221 may also be provided. One or more media drive adapters 242 can be provided for connecting the media drives, for example the optical disk drive 214, the floppy disk drive 216 and hard disk drive 219, to the bus structure 238. One or more telecommunications adapters 244 can be provided for connecting the computer system to one or more networks or to other computer systems or devices.

In operation the processor 230 runs computer software by executing computer program instructions and operating on data that may be stored in one or more of the read only memory 232, random access memory 234 the hard disk drive 219, a floppy disk in the floppy disk drive 216 and an optical disc, for example a compact disc (CD) or digital versatile disc (DVD), in the optical disc drive or dynamically loaded via adapter 244. The results of the processing performed may be displayed to a user via the display adapter 236 and display device 218. User inputs for controlling the operation of the computer system 200 may be received via the user-input device adapters 240 from the user-input devices.

Computer software comprising data files and executable files or computer programs for implementing various functions or conveying various information can be written in a variety of different computer languages and can be supplied on carrier media. Software comprising a program or program element may be supplied on one or more CDs, DVDs and/or floppy disks and then stored on a hard disk, for example. Software may also be embodied as an electronic signal supplied on a

telecommunications medium, for example over a telecommunications network. Examples of suitable carrier media include one or more selected from: a radio frequency signal, an optical signal, an electronic signal, a magnetic disk or tape, solid state memory, an optical disk, a magneto-optical disk, a compact disk and a digital versatile disk.

It will be appreciated that the architecture of a computer system could vary considerably and Figure 22 is only one example.

10 In the present example computer software configured to provide the database is stored on the computer system.

## REFERENCES

- 5 [1] 'Substructure search of chemical structure files'; pp157-181, and 'Chemical structure search systems and services'; pp 182-202, in communication, storage and retrieval of chemical information, Ash J., Chubb P., Welford S., Willet P. (Eds). Ellis Horwood, Chichester, 1985.
- 10 [2] Barnard J.M.; 'Structure representation and searching'; pp 9-56, in Chemical Structure Systems, Ash J.E., Warr W.A., Willet P.(Eds), Ellis Horwood, Chichester, 1991.
- [3] Mooers C.N.; 'Zatocoding applied to mechanical organization of knowledge'; Amer. Doc., 2, 20-32, Jan 1951.
- [4] Mooers C. N.; 'Zatocoding and developments in information retrieval'; ASLIB Proceedings, 8(1), 3-22, Feb 1956.
- 15 [5] J G Vinter: Journal of Computer-Aided Molecular Design: volume 8 (1994) pages 653-668.
- [6] J G Vinter and K I Trollope: Journal of Computer-Aided Molecular Design: volume 9 (1995) pages 297-307.

## ANNEX A - Packages

### PACK CBMD REG

5 This package contains procedures which are concerned with the registration and deletion of molecules and conformations in the database. The following is a list of the public interfaces

#### Procedure DELETE\_ALLOBJECTS

10     **In Parameters:** None  
      **Out Parameters:** None  
      **Return Parameters:** None  
      **Internal Implementation Language:** SQL  
15     **Description:** Deletes all molecules and their conformations from the database and resets the ObjectID\_Seq back to the starting number 1.

#### Procedure DELETE\_OBJECT

**In Parameters:** OBJECTID\_IN, The objectid of the molecule to be deleted  
20     **Out Parameters:** None  
      **Return Parameters:** None  
      **Internal Implementation Language:** SQL  
      **Description:** Deletes the specified molecules and its conformations from the database.

25

#### Procedure DELETE\_STRUCTURE

**In Parameters:**  
          OBJECTID\_IN, The ObjectID of the conformation to be deleted  
30     STRUCTSEQNUM\_IN, the particular conformation to be deleted  
      **Out Parameters:** None  
      **Return Parameters:** None  
      **Internal Implementation Language:** SQL  
35     **Description:** Deletes the specified conformation of the specified molecule from the database.

#### Function REGISTER\_OBJECT

40     **In Parameters:**  
      NAME\_IN, The name for the Molecule to be registered  
      DESCRIPTION\_IN, The Description for the Molecule to be registered  
      TYPENAME\_IN, The Type for the Molecule to be registered  
      SOURCENAME\_IN, The Source for the Molecule to be registered

MOLFORMULA\_IN, The Molecular Formula for the Molecule to be registered  
MOLWEIGHT\_IN, The Molecular Weight for the Molecule to be registered  
5 NUMSTRUCTURES\_IN, The Number of conformations associated with the Molecule to be registered  
MAXENERGY\_IN, The Maximum Energy of a conformation for the Molecule to be registered  
MINENERGY\_IN, The Minimum Energy of a conformation for the  
10 Molecule to be registered  
IMPORTFILE\_IN, The File the Molecule is being imported from.

**Out Parameters:** None

**Return Parameters:** NUMBER

15 **Internal Implementation Language:** SQL

**Description:** Uses the information supplied to register details into the Objects table and returns the ObjectID for the registered molecule.

20 **Procedure REGISTER\_SOURCE**

**In Parameters:**

NAME\_IN, The name of the Source to be registered

DESCRIPTION\_IN, The description of the Source to be  
registered

25

**Out Parameters:** None

**Return Parameters:** None

**Internal Implementation Language:** SQL

**Description:** Registers an entry into the source\_dict table

30

**Procedure REGISTER\_TYPE**

**In Parameters:**

NAME\_IN, The name of the Type to be registered

35 DESCRIPTION\_IN, The description of the Type to be  
registered

**Out Parameters:** None

**Return Parameters:** None

**Internal Implementation Language:** SQL

40 **Description:** Registers an entry into the Type\_dict table

**Procedure REGISTER\_STRUCTURE**

**In Parameters:**

OBJECTID\_IN, The ObjectID of the parent molecule.

45 ENERGY\_IN, The Energy of this particular conformation.

**Out Parameters:** None

**Return Parameters:** None

**Internal Implementation Language:** SQL

**Description:** Registers an entry into the structures table and adds an empty

50 BLOB for structure ready to receive the data.



## PACK CBMD CHEM

This package is concerned with searching the fingerprints.

5

### **Procedure IMPORT**

#### **In Parameters:**

szFileName, The FileName to be imported,  
nFileType, The FileType (Format) of the File (XED .dat file or MDL  
10 .mol file)

szTypeName, The Type which the molecule and its conformations are  
to registered against

The szSourceName, The Source which the molecule and its  
conformations are to registered against

15

**Out Parameters:** None

**Return Parameters:** NONE

**Internal Implementation Language:** JAVA

**Description:** This procedure opens the specified file and reads the contents.  
Entries are registered in to the Objects and Structures tables along with data  
20 read from the file or calculated from the structure. The structures /  
conformations read from the file are converted to Java Objects and stored as  
BLOBs. The fieldprints are created and registered in to the fieldprints table  
ready for searching.

25

### **Procedure DELETERESULTS**

#### **In Parameters:**

szViewName, The name of the view related to the results to be deleted.

**Out Parameters:** None

30

**Return Parameters:** NONE

**Internal Implementation Language:** JAVA

**Description:** Deletes the related results view and the results table for the  
specified name.

35

### **Procedure CREATEINDEX**

#### **In Parameters:**

nIndexLevel, The level of the index to be recreated

**Out Parameters:** None

**Return Parameters:** NONE

40

**Internal Implementation Language:** JAVA

**Description:** Deletes the current fieldprint table entry for the specified level  
and recreates it from the field patterns stored in the structures table.

**Function GETMDLSTRING**

**In Parameters:**

5       nObjectID, The objected of the required molecule  
      nConfNum, The conformation number of the specified  
      molecule  
      nAtoms, Include atoms 1 = yes 0 = no  
      nFields, Include field points 1 = yes 0 = no  
10      nNeg, Include negative field points 1 = yes 0 = no  
      nPos, Include positive field points 1 = yes 0 = no  
      nSur, Include surface field points 1 = yes 0 = no  
      nSca, Include scaffold field points 1 = yes 0 = no  
      nHydrogens, Include hydrogen atoms 1 = yes 0 = no

**Out Parameters: None**

15      **Return Parameters:** String.

**Internal Implementation Language:** JAVA

**Description:** Returns an MDL mol file string of the specified molecule with the specified detail.

20      **Procedure WRITEXEDFILE**

**In Parameters:**

      szFileName, The filename which the structure should be  
      wriiten to.  
25      nObjectID, The objected of the required molecule  
      nConfNum, The conformation number of the specified  
      molecule  
      nAppend, Append the file or overwrite, 1 = Append 0 =  
      overwrite

**Out Parameters: None**

30      **Return Parameters:** NONE

**Internal Implementation Language:** JAVA

**Description:** Writes the specified structure and field points to a file in the XED format.

35      **Function EXACTPATTERN**

**In Parameters:**

      nObjectID, The objected of the required molecule  
      nConfNumber, The conformation number of the specified  
40      molecule  
      nDistance, Include Distance Information 1= Yes 0 = No  
      nEnergy, Include Energy Information 1= Yes 0 = No  
      nFieldCount, Include Distance Information 1= Yes 0 = No  
      nLevel, The precision level of the Fieldprints to be used  
      nNeg, Include Negative field points 1= Yes 0 = No  
45      nPos, Include Positive field points 1= Yes 0 = No  
      nSur, Include Surface field points 1= Yes 0 = No  
      nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters: None**

50      **Return Parameters:** The name of the view containing the results for the  
      query

**Internal Implementation Language:** JAVA

**Description:** Carries out an Exact pattern comparison over the fieldprints of the required precision level. The query fieldprint is calculated on the fly from the specified molecule conformation from the database, incorporating the specified information.

5

**Function EXACTPATTERN**

**In Parameters:**

10        szMolString, Xed file format of the field pattern to be  
         searched for  
         nDistance, Include Distance Information 1= Yes 0 = No  
         nEnergy, Include Energy Information 1= Yes 0 = No  
         nFieldCount, Include Distance Information 1= Yes 0 = No  
         nLevel, The precision level of the Fieldprints to be used  
15        nNeg, Include Negative field points 1= Yes 0 = No  
         nPos, Include Positive field points 1= Yes 0 = No  
         nSur, Include Surface field points 1= Yes 0 = No  
         nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

20        **Return Parameters:** The name of the view containing the results for the  
query

**Internal Implementation Language:** JAVA

25        **Description:** Carries out an exact pattern comparison over the fieldprints of  
         the required precision level. The query fieldprint is calculated on the fly from  
         the structure string specified and incorporating the required information.

**Function SUBPATTERN**

**In Parameters:**

30        nObjectID, The objected of the required molecule  
         nConfNumber, The conformation number of the specified  
         molecule  
         nDistance, Include Distance Information 1= Yes 0 = No  
         nEnergy, Include Energy Information 1= Yes 0 = No  
35        nFieldCount, Include Distance Information 1= Yes 0 = No  
         nLevel, The precision level of the Fieldprints to be used  
         nNeg, Include Negative field points 1= Yes 0 = No  
         nPos, Include Positive field points 1= Yes 0 = No  
         nSur, Include Surface field points 1= Yes 0 = No  
40        nSca, Include Scaffold field points 1= Yes 0 = No

40        **Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the  
query

**Internal Implementation Language:** JAVA

45        **Description:** Carries out a sub pattern comparison over the fieldprints of the  
         required precision level. The query fieldprint is calculated on the fly from the  
         specified molecule conformation from the database, incorporating the  
         specified information.

**Function SUBPATTERN**

50        **In Parameters:**

szMolString, Xed file format of the field pattern to be  
searched for  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
5 nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
10 nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the  
query

**Internal Implementation Language:** JAVA

15 **Description:** Carries out a sub pattern comparison over the fieldprints of the  
required precision level. The query fieldprint is calculated on the fly from the  
structure string specified and incorporating the required information.

#### Function SUPERPATTERN

20 **In Parameters:**

nObjectID, The objected of the required molecule  
nConfNumber, The conformation number of the specified  
molecule  
25 nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
30 nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the  
query

35 **Internal Implementation Language:** JAVA

**Description:** Carries out a super pattern comparison over the fieldprints of the  
required precision level. The query fieldprint is calculated on the fly from the  
specified molecule conformation from the database, incorporating the  
specified information.

40

#### Function SUPERPATTERN

**In Parameters:**

szMolString, Xed file format of the field pattern to be  
searched for  
45 nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
50 nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

5 **Description:** Carries out a super pattern comparison over the fieldprints of the required precision level. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

#### Function EUCLIDIANDIST

10 **In Parameters:**

nObjectID, The objected of the required molecule  
nConfNumber, The conformation number of the specified molecule  
15 nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and database entry  
20 nMinSim, Allowed Minimum similarity between query and database entry  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
25 nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

30 **Description:** Carries out a Euclidean distance comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

35

#### Function EUCLIDIANDIST

**In Parameters:**

szMolString, Xed file format of the field pattern to be searched for  
40 nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
45 nMaxSim, Allowed Maximum similarity between query and database entry  
nMinSim, Allowed Minimum similarity between query and database entry  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
50 nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

5 **Description:** Carries out a Euclidean distance comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

10

### Function SIMTANIMOTO

#### In Parameters:

15 nObjectID, The object of the required molecule  
nConfNumber, The conformation number of the specified molecule  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
20 nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and database entry  
nMinSim, Allowed Minimum similarity between query and database entry  
25 nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

30 **Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

35 **Description:** Carries out a Tanimoto Similarity comparison over the fieldprints of the required precision level. Results in the Max – Min allowed similarity range are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

### Function SIMTANIMOTO

#### In Parameters:

40 szMolString, Xed file format of the field pattern to be searched for  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
45 nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and database entry  
nMinSim, Allowed Minimum similarity between query and database entry  
50 nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No

nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

5       **Internal Implementation Language:** JAVA

**Description:** Carries out a Tanimoto Similarity comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

10

### Function DICESIM

#### **In Parameters:**

15       nObjectID, The objected of the required molecule  
      nConfNumber, The conformation number of the specified molecule  
      nDistance, Include Distance Information 1= Yes 0 = No  
20       nEnergy, Include Energy Information 1= Yes 0 = No  
      nFieldCount, Include Distance Information 1= Yes 0 = No  
      nLevel, The precision level of the Fieldprints to be used  
      nMaxSim, Allowed Maximum similarity between query and database entry  
25       nMinSim, Allowed Minimum similarity between query and database entry  
      nNeg, Include Negative field points 1= Yes 0 = No  
      nPos, Include Positive field points 1= Yes 0 = No  
      nSur, Include Surface field points 1= Yes 0 = No  
      nSca, Include Scaffold field points 1= Yes 0 = No

30       **Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

35       **Description:** Carries out a Dice Similarity comparison over the fieldprints of the required precision level. Results in the Max – Min allowed similarity range are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

### Function DICESIM

40       **In Parameters:**

      szMolString, Xed file format of the field pattern to be searched for  
      nDistance, Include Distance Information 1= Yes 0 = No  
45       nEnergy, Include Energy Information 1= Yes 0 = No  
      nFieldCount, Include Distance Information 1= Yes 0 = No  
      nLevel, The precision level of the Fieldprints to be used  
      nMaxSim, Allowed Maximum similarity between query and database entry  
50       nMinSim, Allowed Minimum similarity between query and database entry  
      nNeg, Include Negative field points 1= Yes 0 = No  
      nPos, Include Positive field points 1= Yes 0 = No

nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the

5 query

**Internal Implementation Language:** JAVA

**Description:** Carries out a Dice Similarity comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

10

### Function SIMTVERSKY

#### In Parameters:

15 nObjectID, The objected of the required molecule  
nConfNumber, The conformation number of the specified molecule  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
20 nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and database entry  
nMinSim, Allowed Minimum similarity between query and database entry  
25 nTrgRatio, The weighting for the target field pattern  
nQryRatio, The weighting for the query field pattern  
nNeg, Include Negative field points. 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
30 nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the

query

35

**Internal Implementation Language:** JAVA

**Description:** Carries out a Tversky Similarity comparison over the fieldprints of the required precision level. Results in the Max – Min allowed similarity range are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

40

### Function SIMTVERSKY

#### In Parameters:

45 szMolString, Xed file format of the field pattern to be searched for  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
50 nMaxSim, Allowed Maximum similarity between query and database entry



nMinSim, Allowed Minimum similarity between query and database entry  
nTrgRatio, The weighting for the target field pattern  
nQryRatio, The weighting for the query field pattern  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

**Description:** Carries out a Tversky Similarity comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

#### Function STREETCARDIST

**In Parameters:**

nObjectID, The object of the required molecule  
nConfNumber, The conformation number of the specified molecule  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and database entry  
nMinSim, Allowed Minimum similarity between query and database entry  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the query

**Internal Implementation Language:** JAVA

**Description:** Carries out a StreetCar Distance comparison over the fieldprints of the required precision level. Results in the Max – Min allowed similarity range are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

#### Function STREETCARDIST

**In Parameters:**

szMolString, Xed file format of the field pattern to be searched for

nDistance, Include Distance Information 1= Yes 0 = No  
 nEnergy, Include Energy Information 1= Yes 0 = No  
 nFieldCount, Include Distance Information 1= Yes 0 = No  
 nLevel, The precision level of the Fieldprints to be used  
 nMaxSim, Allowed Maximum similarity between query and  
 database entry  
 nMinSim, Allowed Minimum similarity between query and  
 database entry  
 nNeg, Include Negative field points 1= Yes 0 = No  
 nPos, Include Positive field points 1= Yes 0 = No  
 nSur, Include Surface field points 1= Yes 0 = No  
 nSca, Include Scaffold field points 1= Yes 0 = No

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the

15 query

**Internal Implementation Language:** JAVA

**Description:** Carries out a StreetCar Distance comparison over the fieldprints of the required precision level. Results in the range Max – Min allowed similarity are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

20

#### Function SUBPATTERNSIM

25

**In Parameters:**

nObjectID, The objected of the required molecule  
 nConfNumber, The conformation number of the specified  
 molecule  
 nDistance, Include Distance Information 1= Yes 0 = No  
 nEnergy, Include Energy Information 1= Yes 0 = No  
 nFieldCount, Include Distance Information 1= Yes 0 = No  
 nLevel, The precision level of the Fieldprints to be used  
 nMaxSim, Allowed Maximum similarity between query and  
 database entry  
 nMinSim, Allowed Minimum similarity between query and  
 database entry  
 nNeg, Include Negative field points 1= Yes 0 = No  
 nPos, Include Positive field points 1= Yes 0 = No  
 nSur, Include Surface field points 1= Yes 0 = No  
 nSca, Include Scaffold field points 1= Yes 0 = No

30

35

40

**Out Parameters:** None

**Return Parameters:** The name of the view containing the results for the

query

**Internal Implementation Language:** JAVA

**Description:** Carries out a SubPattern similarity comparison over the fieldprints of the required precision level. Results in the Max – Min allowed similarity range are written to the results table. The query fieldprint is calculated on the fly from the structure string specified and incorporating the required information.

50

#### Function SUBPATTERNSIM

**In Parameters:**

szMolString, Xed file format of the field pattern to be  
searched for  
nDistance, Include Distance Information 1= Yes 0 = No  
5 nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and  
10 database entry  
nMinSim, Allowed Minimum similarity between query and  
database entry  
nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No  
15 **Out Parameters:** None  
**Return Parameters:** The name of the view containing the results for the  
query  
**Internal Implementation Language:** JAVA  
**Description:** Carries out a SubPattern similarity comparison over the  
20 fieldprints of the required precision level. Results in the range Max – Min  
allowed similarity are written to the results table. The query fieldprint is  
calculated on the fly from the structure string specified and incorporating the  
required information.  
25  
**Function SUPERPATTERNSIM**  
**In Parameters:**  
nObjectID, The objected of the required molecule  
30 nConfNumber, The conformation number of the specified  
molecule  
nDistance, Include Distance Information 1= Yes 0 = No  
nEnergy, Include Energy Information 1= Yes 0 = No  
nFieldCount, Include Distance Information 1= Yes 0 = No  
35 nLevel, The precision level of the Fieldprints to be used  
nMaxSim, Allowed Maximum similarity between query and  
database entry  
nMinSim, Allowed Minimum similarity between query and  
database entry  
40 nNeg, Include Negative field points 1= Yes 0 = No  
nPos, Include Positive field points 1= Yes 0 = No  
nSur, Include Surface field points 1= Yes 0 = No  
nSca, Include Scaffold field points 1= Yes 0 = No  
**Out Parameters:** None  
**Return Parameters:** The name of the view containing the results for the  
45 query  
**Internal Implementation Language:** JAVA  
**Description:** Carries out a SuperPattern similarity comparison over the  
fieldprints of the required precision level. Results in the Max – Min allowed  
50 similarity range are written to the results table. The query fieldprint is  
calculated on the fly from the structure string specified and incorporating the  
required information.

## Function SUPERPATTERNSIM

### In Parameters:

5       szMolString, Xed file format of the field pattern to be  
      searched for  
      nDistance, Include Distance Information 1= Yes 0 = No  
      nEnergy, Include Energy Information 1= Yes 0 = No  
      nFieldCount, Include Distance Information 1= Yes 0 = No  
      nLevel, The precision level of the Fieldprints to be used  
10       nMaxSim, Allowed Maximum similarity between query and  
      database entry  
      nMinSim, Allowed Minimum similarity between query and  
      database entry  
      nNeg, Include Negative field points 1= Yes 0 = No  
      nPos, Include Positive field points 1= Yes 0 = No  
15       nSur, Include Surface field points 1= Yes 0 = No  
      nSca, Include Scaffold field points 1= Yes 0 = No

### Out Parameters: None

Return Parameters: The name of the view containing the results for the  
query

20       **Internal Implementation Language: JAVA**

**Description:** Carries out a SuperPattern similarity comparison over the  
      fieldprints of the required precision level. Results in the range Max - Min  
      allowed similarity are written to the results table. The query fieldprint is  
25       calculated on the fly from the structure string specified and incorporating the  
      required information.

## PACK CBMD UTILS

This package contains general utilities used by the other two packages

30

### Procedure CALCULATE\_TIMESTAMP

      In Parameters: None

      Out Parameters: None

      Return Parameters: NUMBER

35

**Internal Implementation Language: SQL**

**Description:** Returns the current timestamp

### Procedure READ\_UTL\_FILE\_DIR

40

      In Parameters: None

      Out Parameters: None

      Return Parameters: VARCHAR2

**Internal Implementation Language: SQL**

45

**Description:** Returns the UTIL\_FILE\_DIR directory

CLAIMS

1. A computer system comprising a database having a plurality of records, wherein each record comprises a field point representation representing field extrema  
5 for a conformation of a chemical structure.
2. The computer system of claim 1, wherein the database includes records for multiple conformations of the same chemical structure.
- 10 3. The computer system of claim 1 or 2, wherein each record further comprises a structural representation of the chemical structure.
4. The computer system of any preceding claim, each record having an index of the field point representation, wherein the index is a searchable representation of the  
15 field point representation.
5. The computer system of claim 4, wherein the index is a string.
6. The computer system of claim 4 or 5, each record having multiple indexes of  
20 the field point representation, wherein the multiple indexes are representations of the field point representation at different precision levels.
7. The computer system of claim 4, 5 or 6, comprising an indexing mechanism for generating an index of a field point representation.
- 25 8. The computer system of claim 7, wherein an index is a string of length n and the indexing mechanism is configured to:
  - (i) generate a numeric identifier from a characteristic of the field point representation;

(ii) generate one or more numbers in a range from 1 to n in dependence on the numeric identifier;

(iii) increment the bins in the string that correspond to the one or more numbers;  
and

5 (iv) optionally repeat (i) to (iii) for another characteristic of the field point representation.

9. The computer system of claim 8, wherein a characteristic of the field point representation includes one or more of:

10 the number of field points of a particular field of the field point representation;  
the particular field and energy of a field point in the field point representation;  
and

the respective energies of and distance between a field point pairing in the field point representation.

15

10. The computer system of claim 8 or 9, wherein the indexing mechanism is configured to take a measurement of a characteristic of the field point representation to generate the numeric identifier.

20 11. The computer system of claim 10, wherein the indexing mechanism is configured to take a measurement of the characteristic of the field point representation at different levels of precision to generate corresponding multiple indexes which represent the field point representation at different precision levels.

25 12. The computer system of any of claims 8, 9, 10 or 11, wherein the indexing mechanism is configured to:

(i) define a plurality of ranges of possible measurement values;

(ii) take a measurement of a characteristic of the field point representation to produce a measurement value;

(iii) assign the measurement value to a range if the measurement value is within the range;

(iv) optionally repeat (ii) and (iii); and

5 (v) use the number of measurement values assigned to the range to generate the numeric identifier.

13. The computer system of claim 12, wherein the indexing mechanism is configured to define ranges of equal width across all ranges.

10 14. The computer system of claim 12, wherein the indexing mechanism is configured to define a range for smaller measurement values with a narrower width than a range for larger measurement values.

15 15. The computer system of any of claims 12, 13 or 14, wherein the indexing mechanism is configured to generate multiple indexes by defining ranges of different widths for different precision levels.

20 16. The computer system of any of claims 8 to 15, wherein the indexing mechanism is configured to generate one or more numbers in a range from 1 to n in dependence on the numeric identifier by using a deterministic function.

17. The computer system of claim 16, wherein the deterministic function is a pseudo-random number generator or a hash function.

25 18. The computer system of any preceding claim, comprising a searching mechanism for searching the database.

19. The computer system of claim 18, wherein the searching mechanism is configured to:

- (i) compare a query index with an index of a field point representation for a record in the database;
- (ii) identify the record as a hit if the comparison satisfies a search criterion; and
- (iii) repeat (i) and (ii) for a plurality of records.

5

20. The computer system of claim 19, wherein the searching mechanism is further configured to:

receive a search query identifying a field point representation;

form the query index by generating an index of the field point representation

10 identified by the search query.

21. The computer system of any of claims 18 to 20, wherein the search mechanism is configured to search by precision level.

15 22. A graphical user interface configured to enable a user to interface with the database of the computer system of any preceding claim.

23. The graphical user interface of claim 22, comprising:

an interface to enable a user to input data to the database; and/or

20 an interface to enable a user to output data from the database; and/or

an interface to enable a user to delete data from the database; and/or

an interface to enable a user to update data in the database; and/or

an interface to enable a user to browse the database; and/or

an interface to enable a user to search the database; and/or

25 an interface to enable a user to displaying search results.

24. The graphical user interface of claim 22 or 23, comprising:

an interface to enable a user to search the database by submitting a search query identifying a field point representation.

30



25. The graphical user interface of any of claims 22 to 24, comprising:  
an interface to enable a user to select an precision level for a search of the  
database.
- 5 26. The computer system of any of claims 1 to 21, further comprising the  
graphical user interface of any of claims 22 to 25.
27. A database for implementation on a computer system, the database configured  
to support a plurality of records, each record comprising a field point representation  
10 representing field extrema for a conformation of a chemical structure.
28. The database of claim 27, configured to support each record having an index  
of the field point representation, wherein the index is a searchable representation of  
the field point representation.
- 15 29. The database of claim 28, wherein the index is a string.
30. The database of any of claims 27 to 29, configured to support each record  
having multiple indexes of the field point representation, wherein the multiple indexes  
20 are representations of the field point representation at different precision levels.
31. The database of any of claims 27 to 29, comprising an indexing mechanism for  
generating an index of a field point representation.
- 25 32. The database of any of claims 27 to 31, comprising a searching mechanism for  
searching the database.
33. Computer software configured to provide the database of any of claims 27 to  
32.

34. A carrier medium carrying computer software configured to provide the database of any of claims 27 to 32.

35. A method of generating an index of a field point representation representing field extrema for a conformation of a chemical structure, wherein the index is a string with n elements, the method comprising:

- (i) generating a numeric identifier from a characteristic of the field point representation;
- (ii) generating one or more numbers in a range from 1 to n in dependence on the numeric identifier;
- (iii) incrementing the string elements that correspond to the one or more numbers; and
- (iv) optionally repeating (i) to (iii) for another characteristic of the field point representation.

36. The method of claim 35, comprising taking a measurement of a characteristic of the field point representation to generate the numeric identifier.

37. The method of claim 36, comprising taking the measurement of the characteristic of the field point representations at different levels of precision to generate corresponding multiple indexes which represent the field point representation at different precision levels.

38. The method of any of claims 35 to 37, comprising

- (i) defining a plurality of ranges of possible measurement values;
- (ii) taking a measurement of a characteristic of the field point representation to produce a measurement value;
- (iii) assigning the measurement value to a range if the measurement value is within the range;
- (iv) optionally repressing (ii) and (iii); and

(v) using the number of measurement values assigned to a range to generate the numeric identifier.

39. The method of claim 38, comprising defining ranges of equal width across all  
5 ranges

40. The method of claim 38, comprising defining a range for smaller measurement values with a narrower width than a range for larger measurement values.

10 41. The method of any of claims 38, 39 or 40, comprising generating multiple indexes by defining ranges of different widths for different precision levels.

42. The method of any of claims 35 to 41, comprising using a deterministic function to generate one or more numbers in a range from 1 to n in dependence on the  
15 numeric identifier.

43. The method of claim 42, wherein the deterministic function is a pseudo-random number generator or a hash function.

20 44. A method of searching a database having a plurality of records, each record comprising a field point representation representing field extrema for a conformation of a chemical structure and having an index of the field point representation, the method comprising:

(i) comparing a query index with an index of a field point representation for a  
25 record in the database;

(ii) identifying the record as a hit if the comparison satisfies a search criterion;

(iii) repeating (i) and (ii) for a plurality of records; and

(iv) outputting a representation of the records identified as a hit.

30 45. The method of claim 44, further comprising:

receiving a search query identifying a field point representation; and  
forming the query index by generating an index of the field point  
representation identified by the search query.

- 5    46.    The method of claim 44 or 45, the method further comprising searching by  
precision level.

ABSTRACT

SEARCHABLE MOLECULAR DATABASE

5 A computer system comprising a database (100) having a plurality of records is  
provided. Each record comprises a field point representation representing field  
extrema for a conformation of a chemical structure. The database may include records  
for multiple conformations of the same chemical structure. Each record can have a  
searchable index of the field point representation. In one embodiment the index is bit  
string. An indexing mechanism for generating an index, a searching mechanism for  
10 searching the database and a graphical user interface to enable a user to interface with  
the database (100) are also provided.

Figure 20

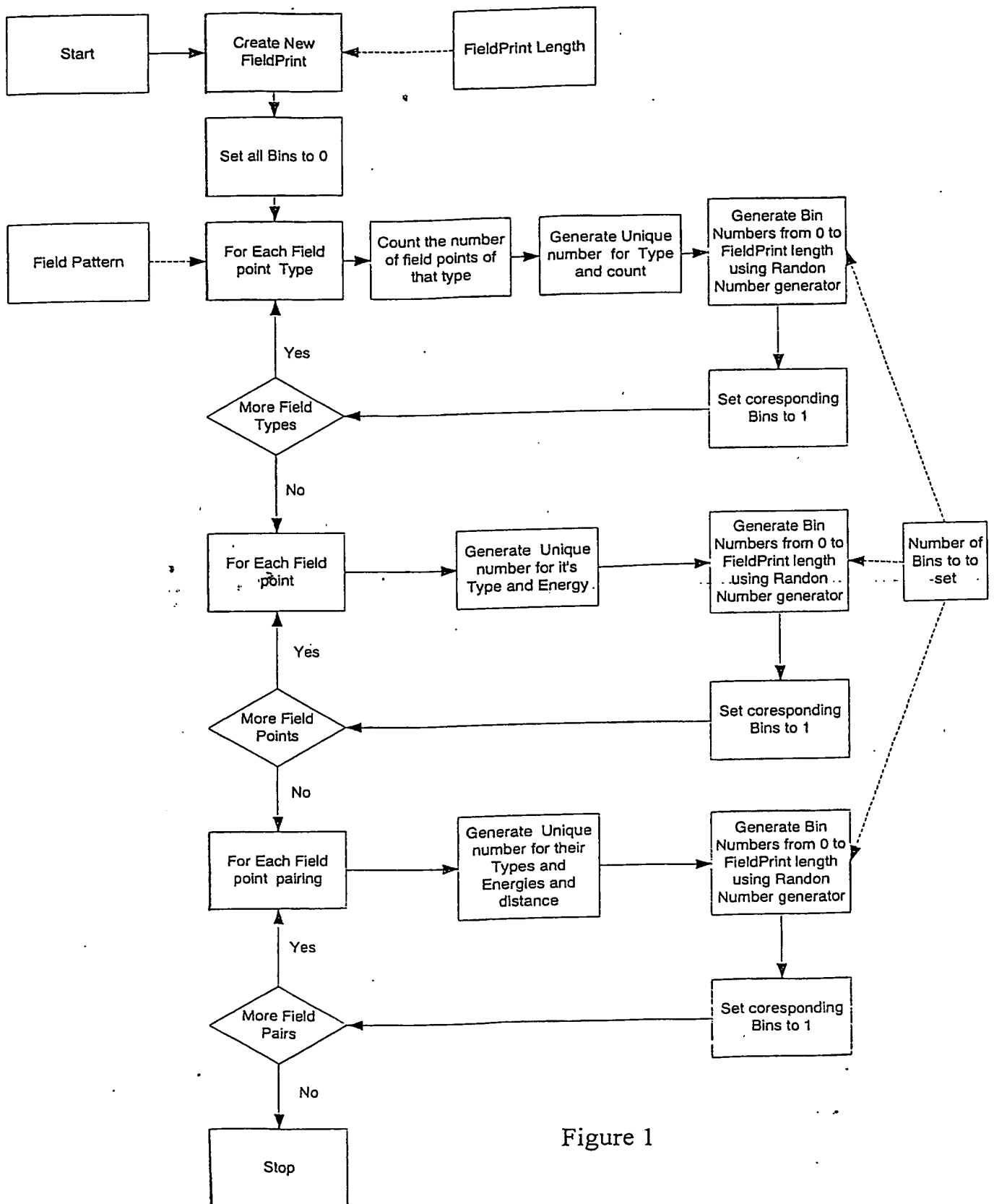


Figure 1

2/16

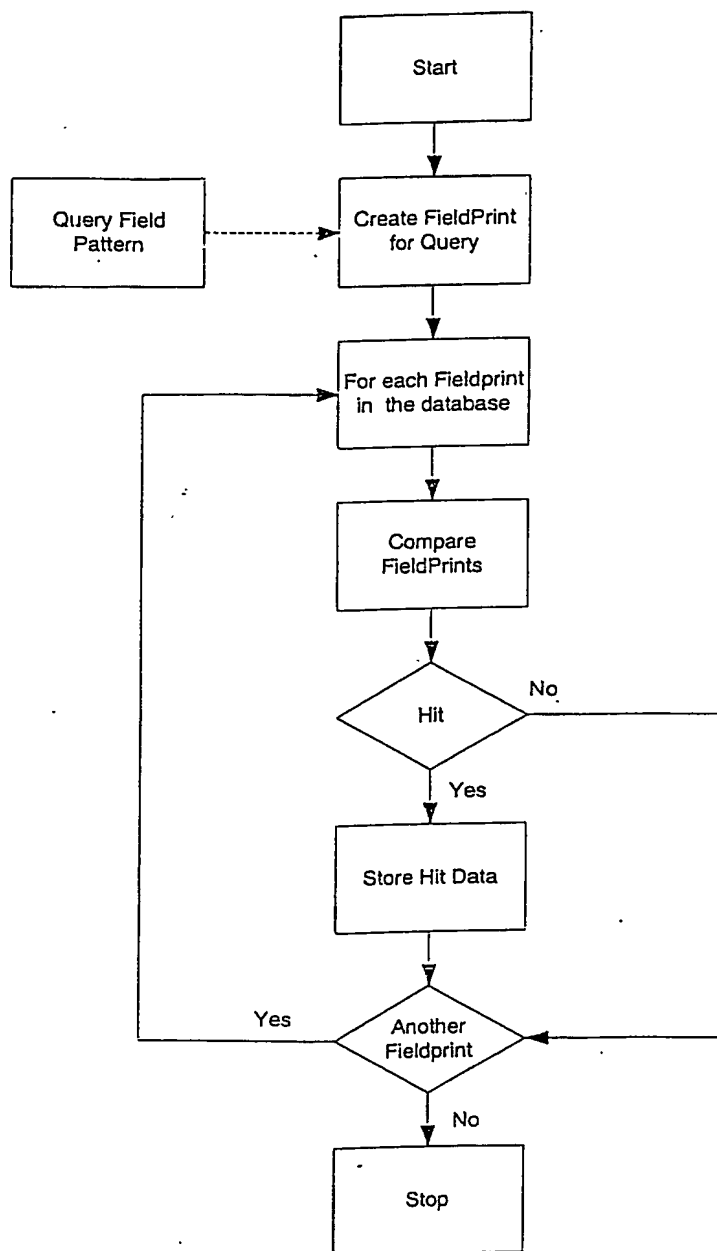


Figure 2

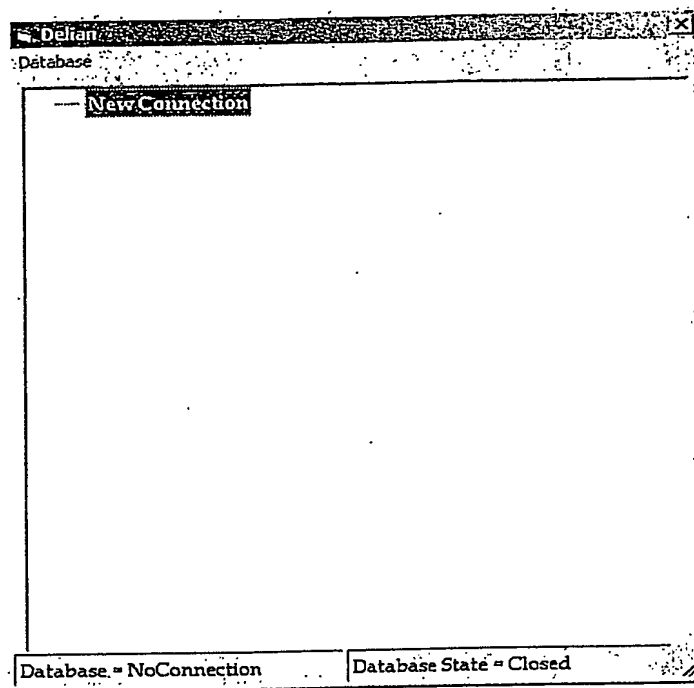


Figure 3

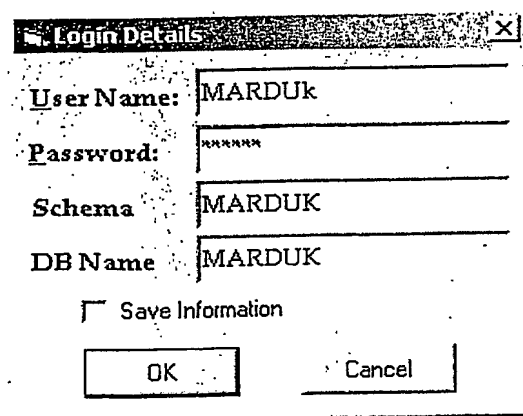


Figure 4



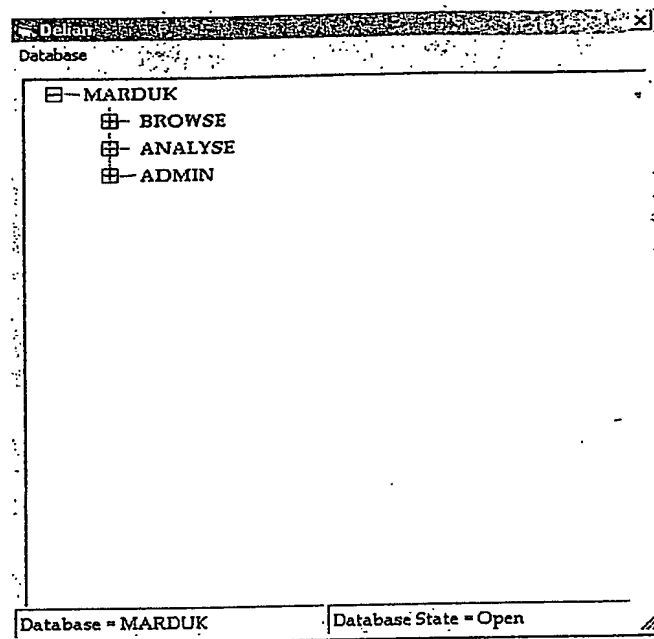


Figure 5

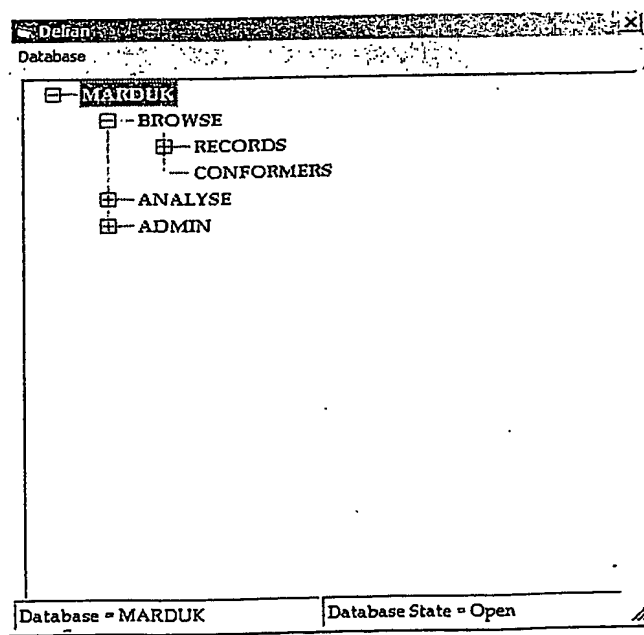


Figure 6

5/16

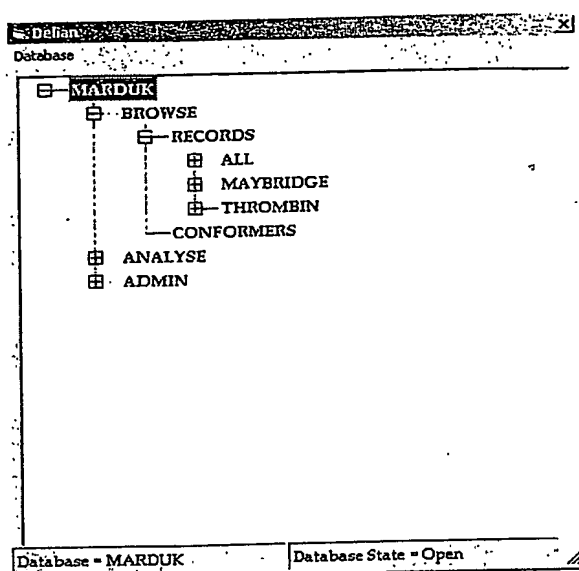


Figure 7

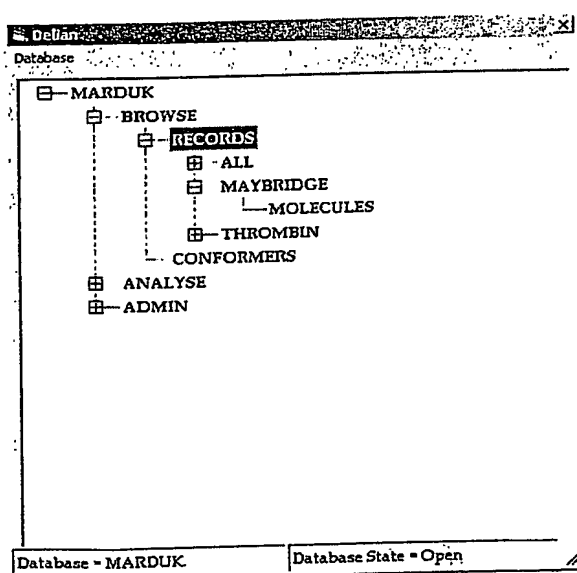


Figure 8

Records for Source: MAYBRIDGE And Type: MOLECULES			
Admin Display			
<b>Information</b>			
Record ID	Name		
50	MayMar01HTS_filt25-50_r10K.1		
Description			
ethyl 2-(1-{2'-[2-(ethoxycarbonyl)ethanehydrazonoyl][1,1'-biphenyl]}-2-y			
Type	Source	Import File	
MOLECULES	MAYBRIDGE	MayMar01HTS_filt25-50_	
Mol Formula	Mol Weight		
0	410.466		
No of Conformers	Max Energy	Min Energy	
3	104.027	98.923	
Browse			
<input type="button" value=" &lt;"/> <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&gt; "/> <input type="button" value="Refresh"/>			
GoTo			
<input type="text"/> <input type="button" value="ID"/> <input type="button" value="Record"/> <input type="button" value="GoTo"/>			
Record 1 of 2001			

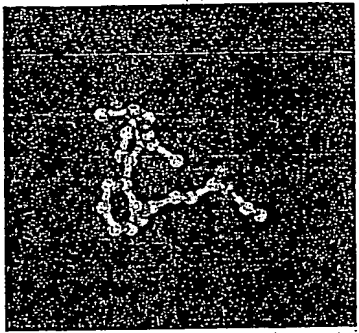
**Sample Structure**  


Figure 9

Records for Source: MAYBRIDGE And Type: MOLECULES			
Admin Display			
<b>Information</b>			
Record ID	Name		
50	MayMar01HTS_filt25-50_r10K.1		
Description			
ethyl 2-(1-{2'-[2-(ethoxycarbonyl)ethanehydrazonoyl][1,1'-biphenyl]}-2-y			
Type	Source	Import File	
MOLECULES	MAYBRIDGE	MayMar01HTS_filt25-50_	
Mol Formula	Mol Weight		
0	410.466		
No of Conformers	Max Energy	Min Energy	
3	104.027	98.923	
Browse			
<input type="button" value=" &lt;"/> <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&gt; "/> <input type="button" value="Refresh"/>			
GoTo			
<input type="text"/> <input type="button" value="ID"/> <input type="button" value="Record"/> <input type="button" value="GoTo"/>			
Record 1 of 2001			

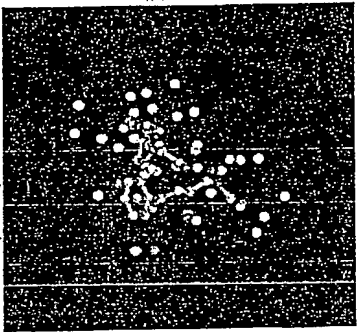
**Sample Structure**  


Figure 10

7/16

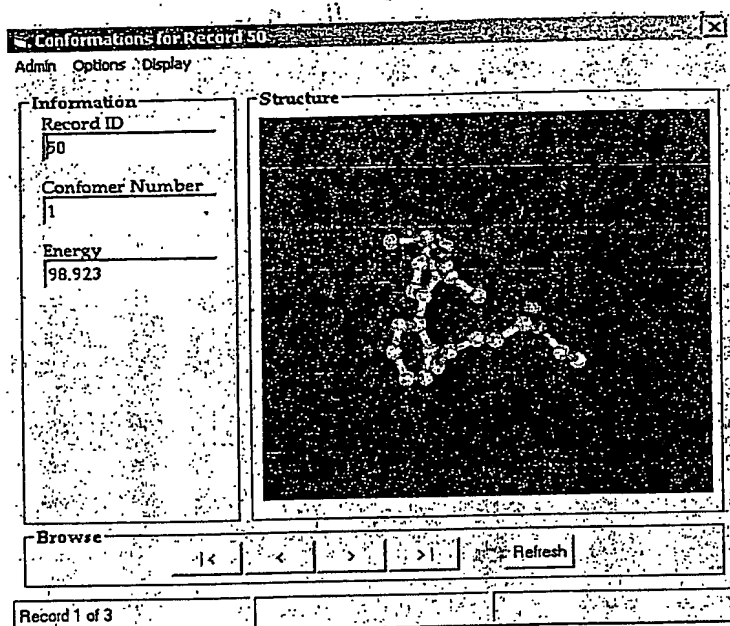


Figure 11

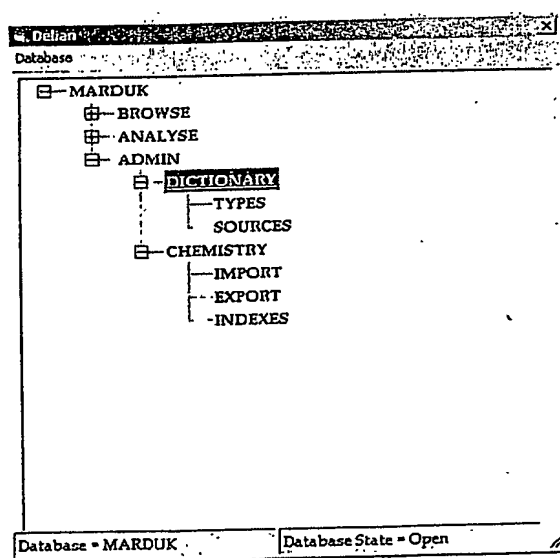


Figure 12

The screenshot shows a window titled "TYPES Dictionary" with a close button (X) in the top right corner. The window is divided into several sections:

- Details:** Contains two text input fields. The "Name" field contains the text "MOLECULES". The "Description" field also contains the text "MOLECULES".
- New Entry:** A vertical stack of three buttons: "New", "Save", and "Cancel".
- Browse:** A horizontal section containing four navigation buttons: "<|", "<", ">", and ">|", followed by a "Refresh" button.
- Record 1 of 2:** A status bar at the bottom of the window.

Figure 13

The screenshot shows a window titled "Import" with a close button (X) in the top right corner. The window contains the following sections:

- Data:** A section header.
- Information:** A text box containing the message: "The files which are to imported should be placed in the UTIL\_FILE\_DIR on the server. See your administrator for more information. Only Xed file types are currently Supported".
- Select the Files for Import:** Contains two radio buttons. The first is "All Files" (selected). The second is "Single File", which is followed by a "FileName" text input field.
- Select the Type and Source:** Contains two dropdown menus labeled "Source" and "Type".
- Import:** Contains an "Import Dir" text input field, an "Import" button, and a checkbox labeled "Continuous Polling".

Figure 14

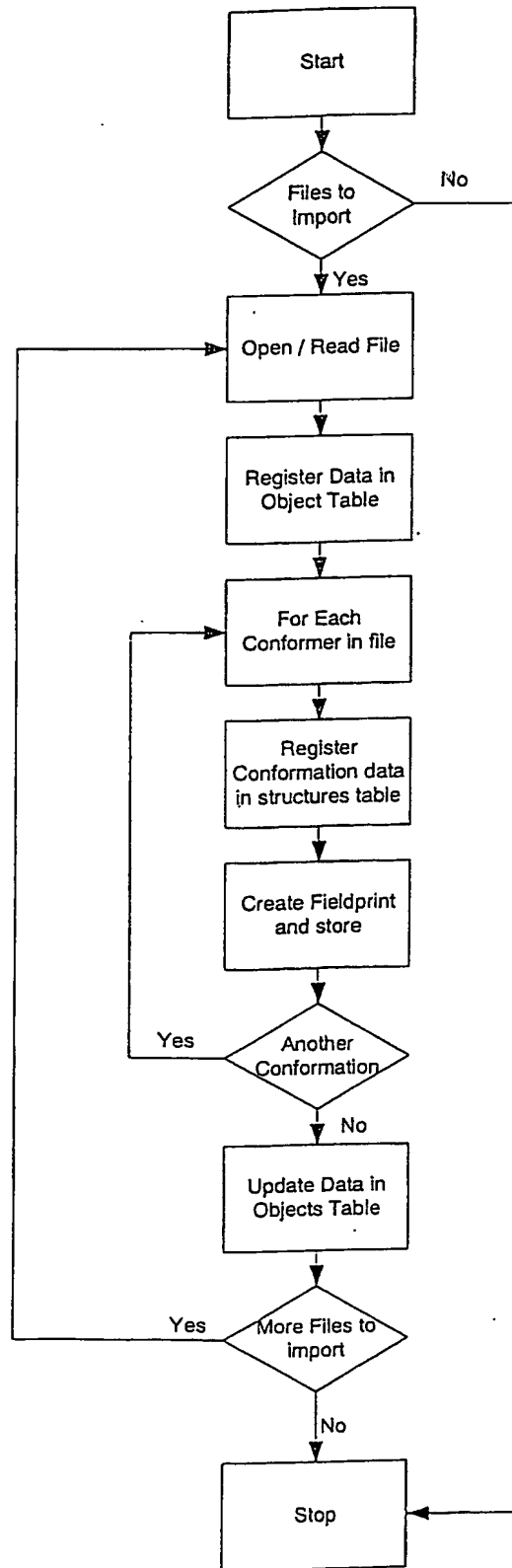


Figure 15

10/16

The image shows a software window titled "EXPORT" with standard Windows window controls (minimize, maximize, close). The window is divided into two main sections. The top section, titled "Select Records to Export", contains two tabs: "Record" (which is selected) and "List". The "Record" tab displays a form for "Record 1" with four dropdown menus labeled "Source", "Type", "Record ID", and "Conformer". The "List" tab is currently empty and has an "Open List" button next to it. The bottom section, titled "Export", contains a text field labeled "FileName" and an "Export" button.

EXPORT

Select Records to Export

☒ Record ☐ List

Record 1

Source

Type

Record ID

Conformer

Export

FileName

Figure 16

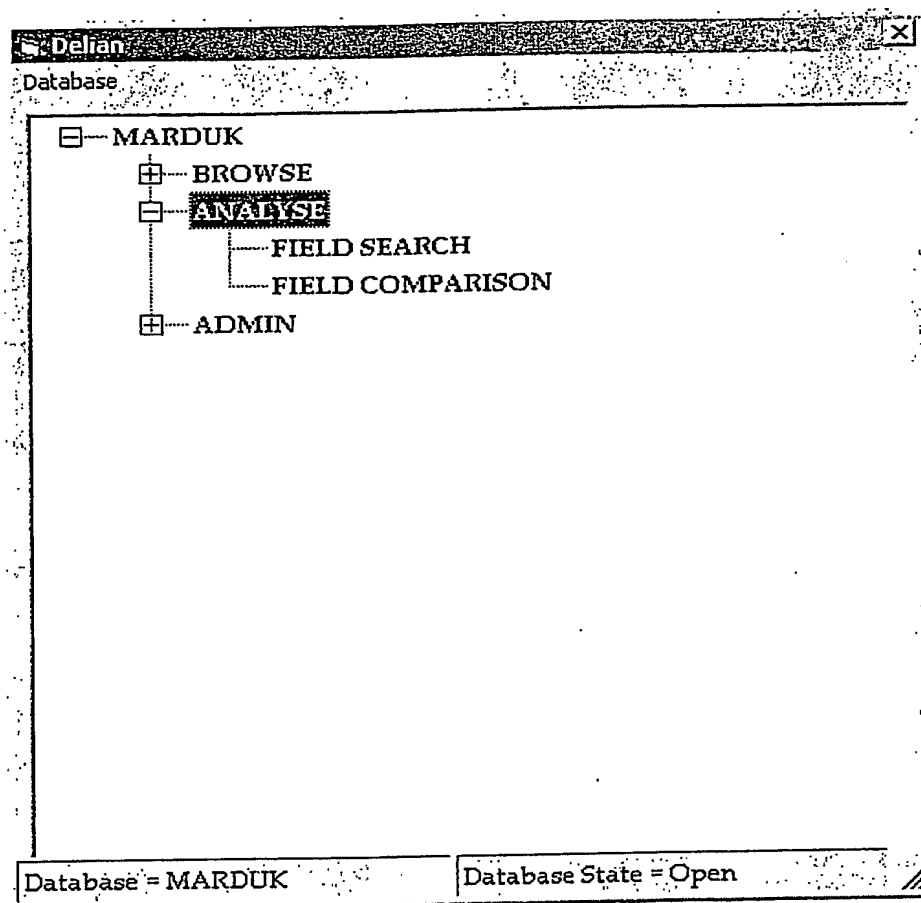


Figure 17



12/16

**Field Search** [X]

File

**Select Data Source**

☒ Database Record Source [ ] Type [ ]

Record ID [ ] Conformer [ ]

☐ From File [ ] Browse [ ]

Conformer [ ]

**Select Search Type**

☒ Exact Pattern ☐ Sub Pattern Sim

☐ Sub Pattern ☐ Super Pattern Sim

☐ Super Pattern ☐ Tanimoto Sim

☐ Euclidian Dist ☐ Dice

☐ StreetCar Dist ☐ Tversky Sim

Qry Ratio [100] Tgt Ratio [100]

Allowed Similarity Values

Max [1.0]

Min [0.5]

**Select Search Index Type**

☒ Pair-Wise Distance ☒ Level 1 ☐ Level 2 ☐ Level 3 ☐ Level 4

☐ Three Way Triangle

**Set Criteria**

Fields ☒ Neg ☒ Pos ☒ Sur ☒ Sca

FPrint ☒ Energy ☒ Field Count

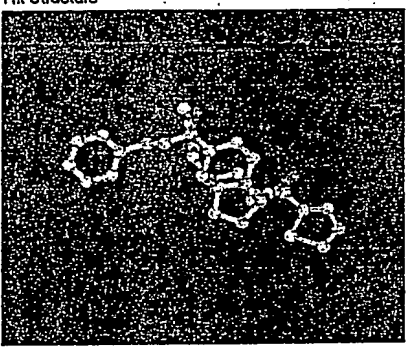
**Search**

Search [ ]

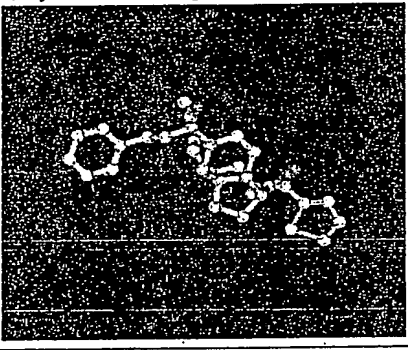
Figure 18

RESULTS			
File Options Display Data Items			
Results Table			
RECORDID	SIMILARITY	SOURCE	TYPE
1	1	THROMBIN	MOLECULES
41	0.908	THROMBIN	MOLECULES
25	0.904	THROMBIN	MOLECULES
40	0.898	THROMBIN	MOLECULES
20	0.892	THROMBIN	MOLECULES
39	0.891	THROMBIN	MOLECULES
42	0.891	THROMBIN	MOLECULES
88	0.886	MAYBRIDGE	MOLECULES
1997	0.884	MAYBRIDGE	MOLECULES
13	0.882	THROMBIN	MOLECULES
833	0.881	MAYBRIDGE	MOLECULES
1915	0.881	MAYBRIDGE	MOLECULES
19	0.88	THROMBIN	MOLECULES
24	0.879	THROMBIN	MOLECULES
1453	0.879	MAYBRIDGE	MOLECULES
81	0.878	MAYBRIDGE	MOLECULES
1406	0.878	MAYBRIDGE	MOLECULES
1167	0.878	MAYBRIDGE	MOLECULES
21	0.877	THROMBIN	MOLECULES
876	0.877	MAYBRIDGE	MOLECULES
2050	0.877	MAYBRIDGE	MOLECULES
23	0.875	THROMBIN	MOLECULES
149	0.875	MAYBRIDGE	MOLECULES
328	0.875	MAYBRIDGE	MOLECULES
496	0.874	MAYBRIDGE	MOLECULES
1305	0.872	MAYBRIDGE	MOLECULES
1345	0.872	MAYBRIDGE	MOLECULES
1659	0.872	MAYBRIDGE	MOLECULES
1569	0.872	MAYBRIDGE	MOLECULES
472	0.871	MAYBRIDGE	MOLECULES

Hit Structure



Query Structure



Total Hits Found = 1446

Current Record = 1

Figure 19

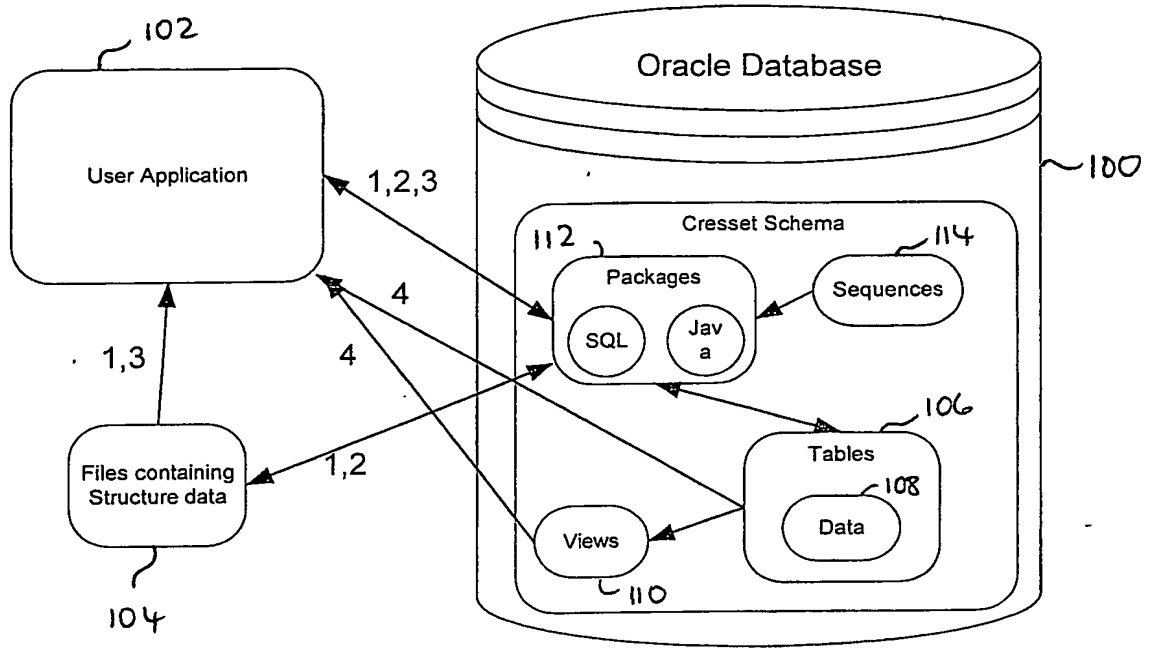


Figure 20

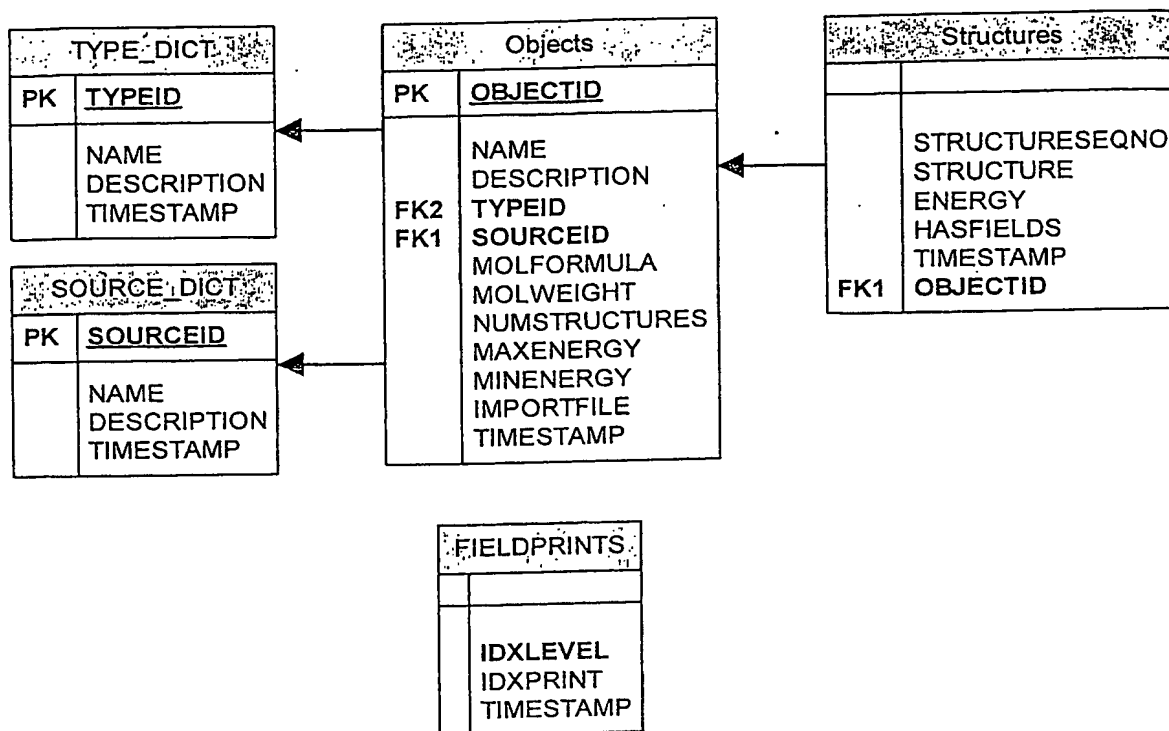


Figure 21

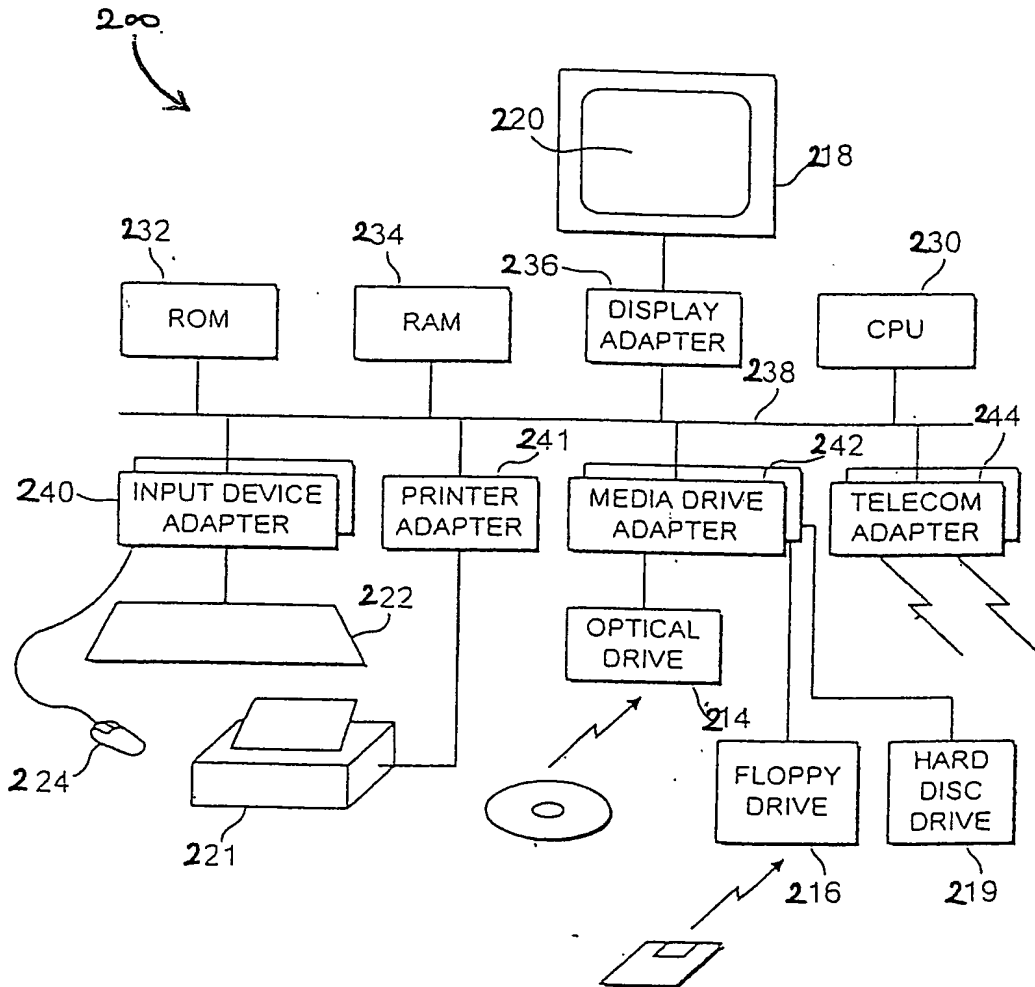


Figure 22

THE PATENT OFFICE  
25 SEP 2003  
Received in Parents  
International Unit

PCT Application  
**GB0303868**



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**